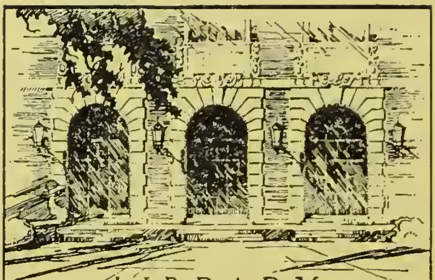


Math

Report No. 266

COO-1469-0077

THE SOLUTION OF SIMULTANEOUS BOOLEAN EQUATIONS

by

HOWARD STEPHEN SCHWEITZER

May 28, 1968

DEPARTMENT OF COMPUTER SCIENCE · UNIVERSITY OF ILLINOIS · URBANA, ILLINOIS

Report No. 266

THE SOLUTION OF SIMULTANEOUS BOOLEAN EQUATIONS

by

HOWARD STEPHEN SCHWEITZER

May 28, 1968

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

ACKNOWLEDGEMENT

The author is deeply indebted to his advisor, Professor W. J. Poppelbaum, whose encouragement, understanding, and support have been invaluable in the completion of this project. The author is also extremely grateful to Professor F. E. Hohn, whose cooperation and guidance were essential in the preparation of this paper.

Thanks are also extended to Mr. M. Payne for many helpful discussions, and to Mrs. Gayle Osterberg for typing the final manuscript.

TABLE OF CONTENTS

1. INTRODUCTION

In many problems, particularly those involving the logical design of switching circuits, the need arises for the solution of simultaneous Boolean equations. Since the application of Boolean algebra to switching theory is a rather recent development, it is not very surprising to find that the amount of work which has been done in this field is relatively sparse.

This paper is expository in nature, and is intended as a survey of the pertinent literature, to date, dealing with this subject. Numerous examples, different from those in the actual papers cited, are given, and in many cases are solved by various methods in order to compare the efficiency of the techniques. (For further comparison, see references [1], [3], [9], [19], and [22] for solutions of the "star-delta-transformations" problem.)

Inasmuch as obtaining the conditions for the existence of solutions verges upon solving the equations, these conditions will not be investigated in great detail. For all problems, the given system of equations will be linear in bivalent variables. Section three presents seven methods for the solution of simultaneous Boolean functional equations, i.e., the determination of dependent variables as functions of independent variables. Particular solutions (i.e., the assignment of a definite value to a bivalent

variable may always be found by simply exhausting all possible combinations of variables; however, for n variables, there are 2^n possibilities, and so methods must be devised to minimize the labor by reducing the number of potential solutions. Section four consists of three such techniques for Boolean equations, and one for pseudo-Boolean equations. Throughout these two sections, all analogous procedures are noted, and the merits of each approach are summarized in the final division of each section, along with possible applications of the methods.

Section five contains topics which are either extensions of the methods of the previous sections, or are related in such a manner as to present an area of future exploration. Finally, the list of references provides an extensive bibliography of a great many papers which would be difficult to locate, due to the diverse and unrelated sources.

2. NOTATION

The notation of the authors of the many works comprising the basis of this paper varies a great deal, as might be suspected. Thus, it is necessary to adopt a uniform notation to be used throughout this paper, in order to avoid confusion. Table 2.1 below contains the notation to be used henceforth, as well as the other various names and representations which may be found elsewhere.

Table 2.1

Symbol	Type of Operation	Name	Other Symbols
$\cap$	Set-theoretical	Intersection (meet)	$\circ$
$\cup$	Set-theoretical	Union (join)	$+$
$\bar{\ }$	Set-theoretical; logical	Complementation; NOT (negation)	$'$
$+$	Logical	OR(inclusive OR; log. sum; disjunction)	$\vee$, $\oplus$
$\cdot$	Logical	AND (logical product; conjunction)	$\wedge$
$\oplus$	Logical	Exclusive OR (ring; ring sum)	$\circ$, $+$

The absence of a symbol between elements will be implicitly understood as logical AND.

(Note: Throughout this paper, the symbol ϕ represents some solution function, as introduced in Section 3.1; it is not to be confused with Φ, the empty set, which was not used in this paper.)

3. GENERAL SOLUTIONS OF BOOLEAN FUNCTIONAL EQUATIONS

3.1. Basic Theorems

Almost all methods of finding solutions to simultaneous Boolean equations are based, either explicitly or implicitly, on the following well-known theorem, quoted here from a paper by R. L. Ashenhurst: [1]

Theorem 3.1. Given any set of relations

$$\begin{aligned} f_1 &= g_1, \\ f_2 &= g_2, \\ &\vdots \\ f_n &= g_n \end{aligned} \tag{3.1}$$

where the f_k and g_k are switching functions of p variables, they can be combined into the single equivalent relation

$$\phi = \prod_{k=1}^{n} (f_k g_k + \bar{f}_k \bar{g}_k) = 1 \tag{3.2}$$

Proof - It is immediately obvious that $\phi = 1$ if and only if each of its factors is equal to one; moreover, this is true only when $f_k = g_k$ (which is exactly the same as the equivalence statement $f_k g_k + \bar{f}_k \bar{g}_k = 1$). Thus, (3.2) is true only when (3.1) is true,

and vice versa. Therefore, it follows that ϕ is a switching function of p variables which has the value one for just those sets of values of the variables which represent solutions of (3.1).

It will be shown in Section 4.1.1 that Theorem 3.1 leads directly to a simple method for determing a particular solution by means of logical algebra.

In many problems, particularly those of circuit synthesis, it is desirable to solve for some of the variables as functions of the remaining variables. The conditions for the existence of solutions to such Boolean functional equations, as well as an indication of the method of solution, are given in another theorem by Ashenhurst [1].

Theorem 3.2. Given the set of relations (3.1), let the p variables consist of r independent variables $x_1, x_2, \ldots, x_r$, and s dependent variables $y_1, y_2, \ldots, y_s$. Then at least one solution of the form

$$\begin{aligned} y_1 &= y_1(x_1, x_2, \ldots, x_r), \\ y_2 &= y_2(x_1, x_2, \ldots, x_r), \\ &\vdots \\ y_s &= y_s(x_1, x_2, \ldots, x_r), \end{aligned} \tag{3.3}$$

exists if and only if all of the $\rho = 2^r$ complete products of the variables $x_1, x_2, \ldots, x_r$ appear in the canonical form of the function

ϕ given by (3.2). If t_j is the number of terms of ϕ containing the product of $x_1, x_2, \ldots, x_r$ corresponding to the integer j, then the number of distinct solutions of the form (3.3) will be

$$\prod_{j=0}^{\rho-1} t_j$$

(Author's note: I have substituted the term "complete product", which is synonymous with "minterm", for "fundamental product", the phrase actually used by Ashenhurst, since the definition of the latter varies among recent authors.)

Proof: If any minterm does not appear in some term of the canonical form of ϕ, assigning values to the independent variables $x_1, x_2, \ldots, x_r$ in such a way that this missing complete product has the value unity will cause (3.1) to have no solution of the form (3.3), regardless of the values of the dependent variables. Since all other minterms are different combinations of the binary variables, they would thus equal zero. Hence the values of ϕ for the combination could not be 1. Therefore, only if all $\rho = 2^r$ minterms appear in the canonical form can the system of equations have a solution. Furthermore, if all products do appear, the $\prod_{j=0}^{\rho-1} t_j$ solutions can be constructed

by selecting ρ terms of ϕ, one containing each product of $x_1, x_2, \ldots, x_r$, and assigning the coefficients in the disjunctive normal form of the functions $y_k(x_1, x_2, \ldots, x_r)$ as 0 or 1, depending on whether the variable y_k appears primed or unprimed in the corresponding selection term of ϕ.

3.2. Logical Algebraic Algorithm

This algorithm is the direct consequence of theorems 3.1 and 3.2, and the steps described here utilize the simple logical algebraic method of R. L. Ashenhurst.

Problem: Given a system of n simultaneous Boolean equations in the general form

$$f_i(x_j, y_k) = g_i(x_j, y_r) \qquad i = 1, 2, \ldots, n \qquad (3.4)$$

where x_j $(j = 1, 2, \ldots, r)$ are independent variables

y_k $(k = 1, 2, \ldots, s)$ are dependent variables

and $r + s = p$ = total number of variables

Solve for y_k (i.e., Equation 3.3).

This is the basic problem for all methods of Section 3.

Algorithm:

Step 1: Find ϕ, according to (3.2).

Step 2: Form a logical table from ϕ by writing each term as a binary number, where primed variables are written as 0, unprimed variables as 1.

Step 3: Choose $\rho = 2^r$ terms of ϕ, one containing each complete product of $x_1, x_2, \ldots, x_r$. (Remember, there will be $\prod_{j=0}^{\rho-1} t_j$ solutions, according to Theorem 3.2.)

Step 4: Treat this table of ρ terms as a truth table, and form the Boolean function(s) for all y_k.

Example: Solve the system of Boolean equations

$$
\begin{aligned}
f_1(x_1,x_2,y_1,y_2) &\equiv (x_1 + y_1)\bar{x}_2 y_2 + x_2(y_1\bar{y}_2 + \bar{y}_1 y_2) \\
&= \bar{x}_1 x_2 y_2 + (\bar{x}_1 + x_2) y_1 \bar{y}_2 + (\bar{x}_2 + y_2) x_1 \bar{y}_1 \\
&\equiv g_1(x_1,x_2,y_1,y_2) \qquad (3.5.1)
\end{aligned}
$$

$$f_2(x_1,x_2,y_1,y_2) \equiv (x_1\bar{x}_2+\bar{x}_1x_2+\bar{y}_2)y_1+(\bar{x}_1\bar{y}_1+\bar{y}_2)\bar{x}_2+x_1x_2\bar{y}_1y_2$$

$$= (\bar{x}_2+\bar{y}_1)\bar{y}_2+(x_2\bar{y}_1+\bar{x}_2y_1)x_1+\bar{x}_1x_2y_1y_2$$

$$\equiv g_2(x_1,x_2,y_1,y_2) \qquad (3.5.2)$$

<u>Step 1</u>: Find ϕ from (3.2).

$$\begin{aligned}
\phi = \{ & [(x_1+y_1)\bar{x}_2y_2+x_2(y_1\bar{y}_2+\bar{y}_1y_2)]\,[\bar{x}_1x_2y_2+(\bar{x}_1+x_2)y_1\bar{y}_2+(\bar{x}_2+y_2)x_1\bar{y}_1] \\
& + [\overline{(x_1+y_1)\bar{x}_2y_2+x_2(y_1\bar{y}_2+\bar{y}_1y_2)}]\,[\overline{\bar{x}_1x_2y_2+(\bar{x}_1+x_2)y_1\bar{y}_2+(\bar{x}_2+y_2)x_1\bar{y}_1}]\}\cdot \\
& \{[(x_1\bar{x}_2+\bar{x}_1x_2+y_2)y_1+(\bar{x}_1\bar{y}_1+\bar{y}_2)\bar{x}_2+x_1x_2\bar{y}_1y_2]\,[(\bar{x}_2+\bar{y}_1)\bar{y}_2+(x_2\bar{y}_1+\bar{x}_2y_1)x_1 \\
& + \bar{x}_1x_2y_1y_2] + [\overline{(x_1\bar{x}_2+\bar{x}_1x_2+y_2)y_1+(\bar{x}_1\bar{y}_1+\bar{y}_2)\bar{x}_2+x_1x_2\bar{y}_1y_2}\ ]\cdot \\
& [\overline{(\bar{x}_2+\bar{y}_1)\bar{y}_2+(x_2\bar{y}_1+\bar{x}_2y_1)x_1+\bar{x}_1x_2y_1y_2}]\}
\end{aligned} \qquad (3.6)$$

$$\begin{aligned}
\phi = \{ & x_1\bar{x}_2\bar{y}_1y_2+\bar{x}_1x_2y_1\bar{y}_2+x_2y_1\bar{y}_2+\bar{x}_1x_2\bar{y}_1y_2+x_1x_2\bar{y}_1y_2+[\bar{x}_1\bar{y}_1+x_2+\bar{y}_2\ ] \\
& [\bar{x}_2+y_1y_2+\bar{y}_1\bar{y}_2]\,[x_1+\bar{x}_2+\bar{y}_2]\,[x_1\bar{x}_2+\bar{y}_1+y_2]\,[x_2\bar{y}_2+\bar{x}_1+y_1]\}\,.
\end{aligned}$$

$$\{x_1\bar{x}_2y_1\bar{y}_2+x_1\bar{x}_2y_1+\bar{x}_1x_2y_1y_2+\bar{x}_2y_1\bar{y}_2+\bar{x}_1\bar{x}_2\bar{y}_1\bar{y}_2+\bar{x}_2\bar{y}_2$$

$$+\bar{x}_2\bar{y}_1\bar{y}_2+x_1x_2\bar{y}_1y_2+ \ [(x_1x_2+\bar{x}_1\bar{x}_2)y_2+\bar{y}_1] \ [x_2+y_2(x_1+y_1)]$$

$$[\bar{x}_1+\bar{x}_2+y_1+\bar{y}_2] \ [y_2+x_2y_1] \ [\bar{x}_1+x_2y_1+\bar{x}_2\bar{y}_1] \ [x_1+\bar{x}_2+\bar{y}_1+\bar{y}_2]\}$$

$$\phi = \{x_1\bar{x}_2\bar{y}_1y_2+\bar{x}_1x_2y_1\bar{y}_2+x_2y_1\bar{y}_2+\bar{x}_1x_2\bar{y}_1y_2+x_1x_2\bar{y}_1y_2+\bar{x}_1\bar{x}_2\bar{y}_1+\bar{x}_1\bar{x}_2\bar{y}_1\bar{y}_2$$

$$+\bar{x}_1\bar{y}_1\bar{y}_2+\bar{x}_1x_2\bar{y}_1\bar{y}_2+x_1x_2y_1y_2+x_1x_2\bar{y}_1\bar{y}_2+x_2\bar{y}_1\bar{y}_2+x_1\bar{x}_2y_1\bar{y}_2\}\cdot$$

$$\{x_1\bar{x}_2y_1\bar{y}_2+x_1\bar{x}_2y_1+\bar{x}_1x_2y_1y_2+\bar{x}_2y_1\bar{y}_2+\bar{x}_1\bar{x}_2\bar{y}_1\bar{y}_2+\bar{x}_2\bar{y}_2+\bar{x}_2\bar{y}_1\bar{y}_2$$

$$+x_1x_2\bar{y}_1y_2+x_1x_2y_1y_2+\bar{x}_1x_2\bar{y}_1y_2+x_1\bar{x}_2\bar{y}_1y_2+\bar{x}_1\bar{x}_2y_1y_2\}$$

$$\phi = x_1\bar{x}_2\bar{y}_1y_2+\bar{x}_1x_2\bar{y}_1y_2+x_1x_2\bar{y}_1y_2+\bar{x}_1\bar{x}_2\bar{y}_1\bar{y}_2+x_1x_2y_1y_2+x_1\bar{x}_2y_1\bar{y}_2 \qquad (3.7)$$

<u>Step 2</u>: Logical table for ϕ

x_1	x_2	y_1	y_2
0	0	0	0
0	1	0	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Step 3: Choose $\rho = 2^r$ terms. In this case, $r = 2$, so only 4 terms are required for a solution. Furthermore, there will be $\prod_{j=0}^{3} t_j = 1 \cdot 1 \cdot 2 \cdot 2 = 4$ solutions. Thus, we choose

(a)

x_1	x_2	y_1	y_2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

(b)

x_1	x_2	y_1	y_2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	0	1

(c)

x_1	x_2	y_1	y_2
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1

(d)

x_1	x_2	y_1	y_2
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	1

Step 4: Treating these as truth tables yields the solutions:

(a) $y_1 = x_1 x_2$ $\qquad$ $y_2 = x_1 + x_2$

(b) $y_1 = 0$ $\qquad$ $y_2 = x_1 + x_2$

(c) $y_1 = x_1 \qquad y_2 = x_2$

(d) $y_1 = x_1\bar{x}_2 \qquad y_2 = x_2$

The solutions may be checked by simple substitution into the given equations. For example, checking solution (c) yields:

$$f_1(x_1,x_2) = (x_1+x_1)\bar{x}_2x_2+x_2(x_1\bar{x}_2+\bar{x}_1x_2) = \bar{x}_1x_2$$

$$g_1(x_1,x_2) = \bar{x}_1x_2\cdot x_2+(\bar{x}_1+x_2)x_1\bar{x}_2+(\bar{x}_2+x_2)x_1\bar{x}_1 = \bar{x}_1x_2 = f_1(x_1,x_2) \checkmark$$

$$f_2(x_1,x_2) = (x_1\bar{x}_2+\bar{x}_1x_2+\bar{x}_2)x_1+(\bar{x}_1\bar{x}_1+\bar{x}_2)\bar{x}_2+x_1x_2\bar{x}_1x_2 = x_1\bar{x}_2+\bar{x}_1\bar{x}_2=\bar{x}_2$$

$$g_2(x_1,x_2) = (\bar{x}_2+\bar{x}_1)\bar{x}_2+(x_2\bar{x}_1+\bar{x}_2x_1)x_1+\bar{x}_1x_2x_1x_2 = \bar{x}_2+\bar{x}_1\bar{x}_2+x_1\bar{x}_2 = \bar{x}_2$$

$$= f_2(x_1,x_2) \checkmark$$

This method is very laborious if the number of terms in each function is greater than 2, and/or the number of equations is greater than 2.

3.3 "Map" Methods

In order to overcome the tiresome task of logical expansion in the above algorithm, many authors, including Maitra[12], Svoboda[22], Ledley[10], and Nadler[13], employ a Veitch[23] chart, (also called a logical matrix, binary map, and Boolean matrix).

The binary map commonly known as the Veitch chart was first proposed by Marquand in 1881. It employs straight binary distribution, (as opposed to the Gray code in a Karnough map), thus locating the states consecutively, from left to right, and top to bottom. The bars alongside the left and top of the map indicate the rows and columns, respectively, where the variable has the value unity; those without bars have value zero (i.e., the complement of variable has the value unity).

Figure 1 is a four variable map, where the assignment of the variables to the rows and columns is indicated to the right and bottom of the map, respectively. Each cell of the map contains the state index, given as the binary number $x_4x_3x_2x_1$, and the corresponding decimal notation is given in the second map.

All of the following methods are based on Theorem 3.1 (and in some cases, Theorem 3.2), and since 0 may be easily formulated with the aid of a Veitch chart, these methods possess a high degree of similarity.

	$\bar{x}_2\bar{x}_1$	$\bar{x}_2x_1$	$x_2\bar{x}_1$	x_2x_1
$\bar{x}_4\bar{x}_3$	0000	0001	0010	0011
$\bar{x}_4x_3$	0100	0101	0110	0111
$x_4\bar{x}_3$	1000	1001	1010	1011
x_4x_3	1100	1101	1110	1111

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Figure 1.

3.3.1 Svoboda's Algorithm

Although basically identical to Ashenhurst's method, Svoboda's algorithm can solve a large number of equations containing functions of many variables without great difficulty. Svoboda's use of the Veitch chart is basic, and a Karnaugh map may be substituted if desired, with no loss in any way.

The problem is the same as that of Section 3.2: Given the system of Equations (3.4), solve for all solutions of y_k.

Algorithm:

Step 1: Form n maps of Boolean functions

$$\phi_k = f_k g_k + \bar{f}_k \bar{g}_k \qquad (k=1,2,\ldots,n) \qquad (3.8)$$

Each map will have 2^r columns, and 2^s rows, where r and s are the numbers of independent and dependent variables, respectively.

Step 2: Form the "discriminant" ϕ, of the system as the Boolean product of ϕ_k (by finding the intersection of the maps - i.e., logical multiplication of corresponding cells in all maps).

$$\phi = \phi_1 \cdot \phi_2 \cdots \phi_n$$

The "discriminant" ϕ is identical to ϕ in (3.2), and is more simply found graphically than by logical operations. In the map of ϕ, cells containing ones correspond to the terms found by expanding ϕ logically.

Step 3: Calculate the "value" $[\phi]$ of the discriminant as

$$[\phi] = \prod_{j=0}^{\rho-1} t_j = T \qquad (t_j = \text{number of 1's per column})$$

as in Theorem 3.2. The value T is the product of the numbers of 1's in the columns of the maps of ϕ, and gives the number of distinct solutions.

Step 4: Decompose the map of ϕ into all T possible different Boolean components ϕ^t, $t = 1,2,\ldots,T$, where each ϕ^t has only a single 1 per column. $\phi = \phi^1 + \phi^2 + \ldots + \phi^T$ (logical addition for each cell).

This is equivalent to choosing all sets of 2^r complete product terms of ϕ, as in Step 3 of Ashenhurst's method.

Step 5: From each ϕ^t, develop s functions

$$Y_k^t = \phi^t y_k \qquad (k=1,2,\ldots,s)$$

by logical multiplication of the graphical representation of y_k and ϕ^t. This selects those 1's in the ϕ^t-map that are also in the 1-region of the y_k-map.

Step 6: Transform each Y_k^t into y_k^t by filling with 1's every column of Y_k^t containing a 1.

Step 7: Express each y_k^t as a Boolean function by means of the y_k^t maps.

These last three steps are equivalent to the usual method of forming the Boolean functions y_k^t from the truth tables, as in Ashenhurst's last step.

Remarks on the Algorithm

1. ϕ may also be found as $\overline{(\sum_{k=1}^{n} \bar{\phi}_k)}$, where the union of all $\bar{\phi}_k$ is plotted instead of the intersection of all ϕ_k; thus, only a single map is required, and will result in $\bar{\phi}$. This procedure shortens Step 1 and eliminates Step 2.

2. If the positions of the dependent and independent variables were reversed (i.e., rotate the map clockwise 90°), the ϕ^t maps could be treated as truth tables. y_k^t may be found by obtaining the sum of minterms of x's whose columns contain 1's (in Step 4) which appear in corresponding rows of y_k^t. This simple mental procedure eliminates Steps 5 and 6.

Equations (3.5) will be solved now by Svoboda's algorithm, following his steps exactly. The reader will see how this process may be shortened, in the light of Remarks 1 and 2.

Step 1: Form the maps of $f_1, g_1, \phi_1, f_2, g_2, \phi_2$ (Figure 2).

Step 2: Form ϕ (Figure 3).

Step 3: $[\phi] = \prod_{j=0}^{3} t_j = 1\cdot2\cdot1\cdot2 = 4$ solutions

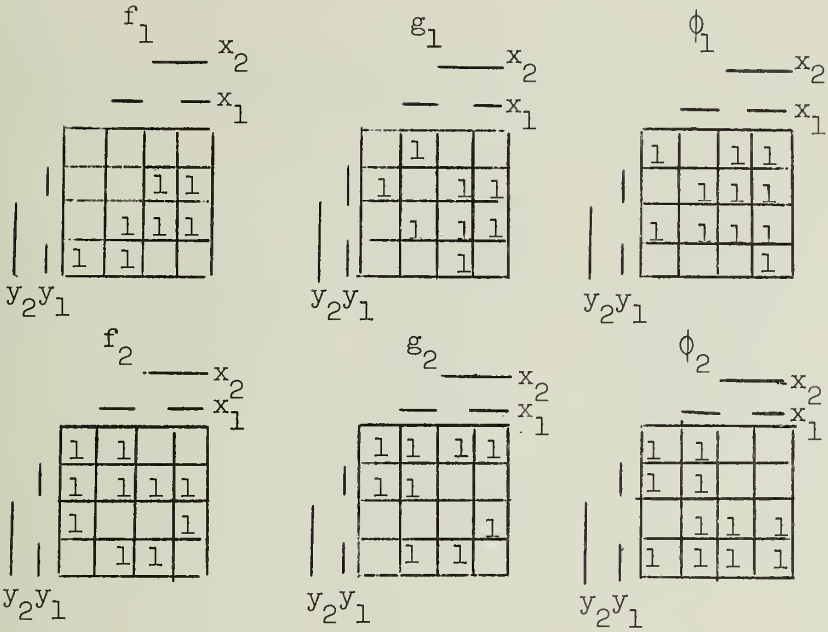


Figure 2.

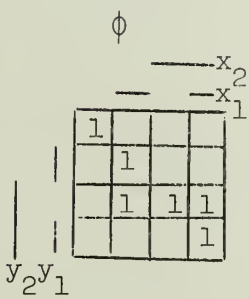


Figure 3.

<u>Step 4</u>: Decompose ϕ into ϕ^t, $t = 1,2,3,4$ (Figure 4).

$$\overset{\displaystyle\phi}{\begin{array}{|c|c|c|c|}\hline 1 & & & \\ \hline & 1 & & \\ \hline & 1 & 1 & 1 \\ \hline & & & 1 \\ \hline \end{array}} \qquad \overset{\displaystyle\phi^1}{\begin{array}{|c|c|c|c|}\hline 1 & & & \\ \hline & 1 & & \\ \hline & & 1 & 1 \\ \hline & & & \\ \hline \end{array}} \qquad \overset{\displaystyle\phi^2}{\begin{array}{|c|c|c|c|}\hline 1 & & & \\ \hline & & & \\ \hline & 1 & 1 & 1 \\ \hline & & & \\ \hline \end{array}} \qquad \overset{\displaystyle\phi^3}{\begin{array}{|c|c|c|c|}\hline 1 & & & \\ \hline & 1 & & \\ \hline & & 1 & \\ \hline & & & 1 \\ \hline \end{array}} \qquad \overset{\displaystyle\phi^4}{\begin{array}{|c|c|c|c|}\hline 1 & & & \\ \hline & & & \\ \hline & 1 & 1 & \\ \hline & & & 1 \\ \hline \end{array}}$$

<u>Figure 4.</u>

<u>Step 5</u>: Find $Y_k^t = \phi^t y_k \qquad k = 1,2,\ldots,s$ (Figure 5).

$$\begin{array}{c} \\ \\ k=1 \\ \\ \\ \end{array}\overset{\displaystyle y_1}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline 1 & 1 & 1 & 1 \\ \hline & & & \\ \hline 1 & 1 & 1 & 1 \\ \hline \end{array}} \qquad \overset{\displaystyle Y_1^1}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & 1 & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}} \qquad \overset{\displaystyle Y_1^2}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}} \qquad \overset{\displaystyle Y_1^3}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & 1 & & \\ \hline & & & \\ \hline & & & 1 \\ \hline \end{array}} \qquad \overset{\displaystyle Y_1^4}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & 1 \\ \hline \end{array}}$$

$$\begin{array}{c} \\ \\ k=2 \\ \\ \\ \end{array}\overset{\displaystyle y_2}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 \\ \hline \end{array}} \qquad \overset{\displaystyle Y_2^1}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & 1 & 1 \\ \hline & & & \\ \hline \end{array}} \qquad \overset{\displaystyle Y_2^2}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & 1 & 1 & 1 \\ \hline & & & \\ \hline \end{array}} \qquad \overset{\displaystyle Y_2^3}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & & 1 & \\ \hline & & & 1 \\ \hline \end{array}} \qquad \overset{\displaystyle Y_2^4}{\begin{array}{|c|c|c|c|}\hline & & & \\ \hline & & & \\ \hline & 1 & 1 & \\ \hline & & & 1 \\ \hline \end{array}}$$

<u>Figure 5.</u>

Step 6: Y_k^t into y_k^t (Figure 6).

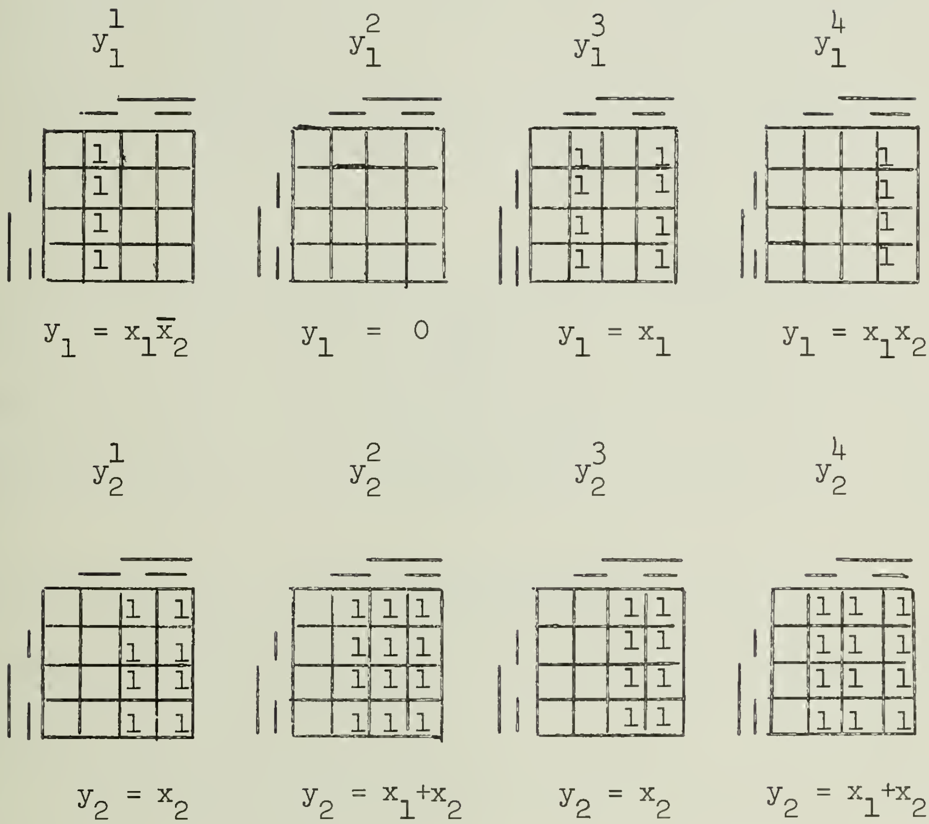


Figure 6.

Step 7: Find y_k^t as a Boolean function.

Singular Cases

Svoboda does add an important case not treated by Ashenhurst. As noted earlier, if $[\varphi] = 0$, i.e., a column of φ has no units, no solution of the form (3.3) exists. However, there does exist a solution of this form which is combined with a constraint of the form $c(x_j) = 1$, which restricts the domain of possible solutions.

If a column of zeros is present, the combination of binary variables that makes the corresponding minterm of x's equal to 1 must be excluded. I.e., the corresponding minterm must always be zero. The complement of the equation is the constraint mentioned above.

If more than one column of zeros are present, the subsequent constraint equations are simply multiplied to obtain the total constraint. Furthermore, the entries of the columns may be treated as don't-cares for simplification of the solution(s).

Example 2: Given the system of Boolean equations,

$$f_1(x_1,x_2,y_1,y_2) \equiv (x_1+y_2)\bar{x}_2+\bar{x}_1y_1y_2+x_1y_1\bar{y}_2+\bar{x}_1x_2\bar{y}_1\bar{y}_2$$

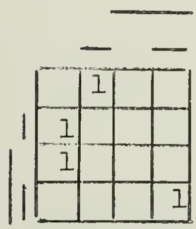
$$= \bar{y}_1+\bar{x}_2y_2 \equiv g_1(x_1,x_2,y_1,y_2)$$

$$f_2(x_1,x_2,y_1,y_2) \equiv x_1x_2+y_1\bar{y}_2+\bar{x}_1\bar{y}_1\bar{y}_2+\bar{x}_2\bar{y}_1y_2$$

$$= \bar{x}_1\bar{x}_2+x_1(x_2+y_1) \equiv g_2(x_1,x_2,y_1,y_2)$$

The reader may verify that the discriminant is Figure 7.

$\phi =$



$[\phi] = 0$

Figure 7.

Steps 4,5, and 6 lead to the y_k^t maps (Figure 8).

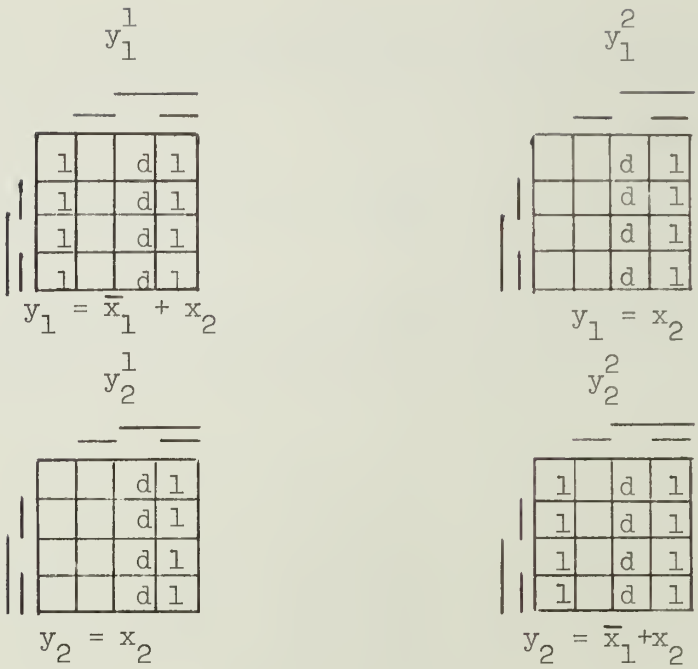


Figure 8.

Constraint: $\overline{\bar{x}_1 x_2} = 0$ or $x_1 + \bar{x}_2 = 1$ (for both solutions).

Check: (Solution 1 in Equation 1) $y_1 = \bar{x}_1 + x_2 \quad y_2 = x_2 \quad x_1 + \bar{x}_2 = 1$

$$f_1 = x_1\bar{x}_2 + x_2\bar{x}_2 + \bar{x}_1(\bar{x}_1 + x_2)x_2 + x_1(\bar{x}_1 + x_2)\bar{x}_2 + \bar{x}_1 x_2(x_1\bar{x}_2)\bar{x}_2$$

$$= (x_1\bar{x}_2 + \bar{x}_1 x_2)(1) = (x_1\bar{x}_2 + \bar{x}_1 x_2)(x_1 + \bar{x}_2) = x_1\bar{x}_2$$

$$g_1 = x_1\bar{x}_2 + \bar{x}_2 x_2 = x_1\bar{x}_2$$

3.3.2 Maitra's Methods

In many problems (particularly those of circuit design), it may be necessary to solve n equations of the form

$$g_k(x_1,x_2,\ldots,x_m,x_{m+1},\ldots,x_r) = f_k(y_1,y_2,\ldots,y_\ell,x_{m+1},\ldots,x_r)$$

$$(k = 1,2,\ldots,n) \qquad (3.9)$$

where the solutions are to have the form

$$\begin{aligned} y_1 &= y_1(x_1,x_2,\ldots,x_m) \\ y_2 &= y_2(x_1,x_2,\ldots,x_m) \\ \vdots\ & \quad \vdots \\ y_\ell &= y_\ell(x_1,x_2,\ldots,x_m) \end{aligned} \qquad (3.10)$$

This is just a special case of (3.4). The situation under consideration is shown in Figure 9.

The general approach to a system of equations like (3.9) is to solve each equation independently, and then find the common solution for all the equations.

If one allows the y's to be given explicitly, considering k = 1 for (3.9), there are, altogether, three basic problem types which may arise:

1. Given the functions f and g explicitly, determine the y_j's.
2. Given the functions f and y_j's explicitly, determine g.
3. Given the functions g and y_j's explicitly, determine f.

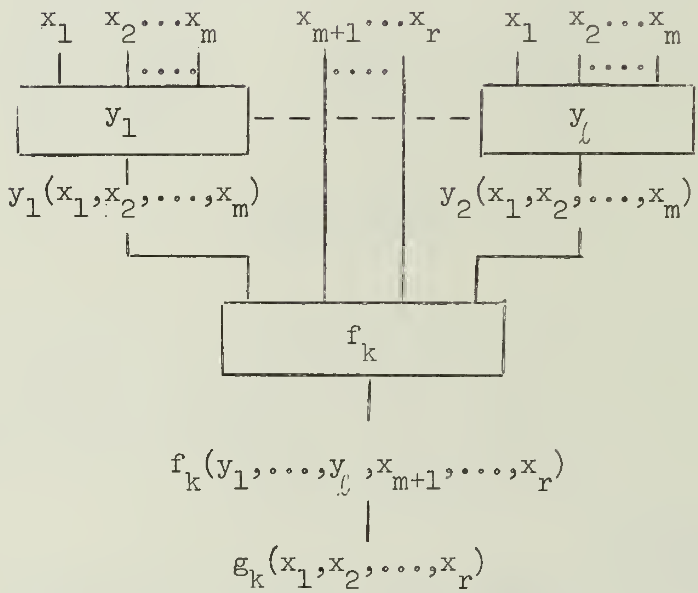


Figure 9.

K. K. Maitra's approach is also based on Theorem 3.1, and is identical in principle to that of Svoboda. However, since this is a special case of the problems previously considered, it has fewer degrees of freedom. Consequently, a modified Veitch chart in tabular form is used to calculate ϕ_k, and to solve the three problems listed above.

Solution for problem of type 1

If k = 1 in (3.9), since the y's are to be expressed in terms of $x_1, x_2, \ldots, x_m$, the single equation f = g must be decomposed into a system of equations so that (3.9) holds for all combinations of the r - m independent variables $x_{m+1}, \ldots, x_r$. Hence, there will result a system of 2^{r-m} equations $f_o = g_o, f_1 = f_1, \ldots, f_{2^{r-m}-1} = g_{2^{r-m}-1}$ to be solved simultaneously for $y_1, y_2, \ldots y_\ell$. Let the index of the resulting functions f_k and g_k be the decimal equivalent of the binary number corresponding to the complete product of $x_{m+1}, \ldots, x_r$ which determines the equation $f_k = g_k$.

Algorithm:

Step 1: Set up a modified Veitch chart for each of the 2^{r-m} equations as follows:

(a) Designate the columns and rows by the binary numbers corresponding to the 2^ℓ and 2^m combinations of the dependent variables

$y_1, y_2, \ldots, y_\ell$ and independent variables $x_1, x_2, \ldots, x_m$, respectively, or their decimal equivalents. (See Remark 2, Svoboda's algorithm, Section 3.3.1.)

(b) List the values of the functions f_k and g_k, corresponding to the minterms of $y_1, y_2, \ldots, y_\ell$ and $x_1, x_2, \ldots, x_m$, along the bottom and right side of the chart, respectively (see Figure 10).

<u>Step 2</u>: Form the ϕ_k chart by "multiplying" the values of f_k and g_k according to the following rules:

$$1 \cdot 1 = 1, \quad 0 \cdot 0 = 1, \quad 1 \cdot 0 = 0, \quad 0 \cdot 1 = 0 \tag{3.11}$$

This is precisely due to the equivalence statement of (3.2), and results in the same maps as those in Step 1 of Svoboda's algorithm.

<u>Step 3</u>: Form the ϕ chart as the Boolean product of the ϕ_k (identical to Svoboda's Step 2).

<u>Step 4</u>: Calculate the number of solutions, as $[\phi] = T$ (identical to Svoboda's Step 3, but now t_j = number of 1's per <u>row</u>).

Maitra does not detail how to obtain the $[\phi]$ solutions as Boolean functions, but obviously the method parallels that of

Svoboda and Ashenhurst. Thus, by decomposing ϕ into ϕ^t charts, and treating each as a truth table, the functions $y_1, y_2, \ldots, y_\ell$ may be found (see Remark 2 in Svoboda's algorithm). Also, if any row has no units, no solution of the form (3.10) exists. However, this may be used to form a solution with a constraint, as in Svoboda's treatment of singular cases.

Example: Given the system of equations

$$f_1(y_1,y_2,x_4,x_5) \equiv y_1y_2\bar{x}_4\bar{x}_5+\bar{y}_1y_2(x_4x_5+\bar{x}_4\bar{x}_5)+x_4\bar{y}_2(x_5y_1+\bar{x}_5\bar{y}_1)$$

$$+x_4y_1(x_5y_2+\bar{x}_5\bar{y}_2)$$

$$= x_3(\bar{x}_4\bar{x}_5+x_4x_5)+\bar{x}_2\bar{x}_3x_4(x_1x_5+\bar{x}_1\bar{x}_5)+x_2\bar{x}_3x_4(\bar{x}_1x_5+x_1\bar{x}_5)+\bar{x}_3x_4x_5\cdot$$

$$(x_1x_2+\bar{x}_1\bar{x}_2)+\bar{x}_3x_4\bar{x}_5(x_1\bar{x}_2+\bar{x}_1x_2) \equiv g_1(x_1,x_2,\ldots,x_5) \qquad (3.12.1)$$

$$f_2(y_1,y_2,x_4,x_5) \equiv y_1y_2(\bar{x}_4+x_5)+\bar{y}_1y_2(\bar{x}_4x_5+x_4\bar{x}_5)+\bar{x}_5\bar{y}_1(x_4\bar{y}_2+\bar{x}_4y_2)$$

$$+ x_5\bar{y}_2(\bar{x}_4\bar{y}_1+x_4y_1)+y_1\bar{y}_2(x_4\bar{x}_5+\bar{x}_4x_5)$$

ϕ_1^0 $\quad y_1y_2$

	$x_1x_2x_3$	00 (0)	01 (1)	10 (2)	11 (3)	g_0
(0)	0 0 0	1	0	1	0	0
(1)	0 0 1	0	1	0	1	1
(2)	0 1 0	1	0	1	0	0
(3)	0 1 1	0	1	0	1	1
(4)	1 0 0	1	0	1	0	0
(5)	1 0 1	0	1	0	1	1
(6)	1 1 0	1	0	1	0	0
(7)	1 1 1	0	1	0	1	1
	f_0	0	1	0	1	

ϕ_1^1

	00	01	10	11	g_1
0	1	1	1	1	0
1	1	1	1	1	0
2	1	1	1	1	0
3	1	1	1	1	0
4	1	1	1	1	0
5	1	1	1	1	0
6	1	1	1	1	0
7	1	1	1	1	0
f_1	0	0	0	0	

Figure 10.

ϕ_1^2

	0	1	2	3	g_2
0	1	0	1	0	1
1	0	1	0	1	0
2	1	0	1	0	1
3	0	1	0	1	0
4	1	0	1	0	1
5	0	1	0	1	0
6	1	0	1	0	1
7	0	1	0	1	0
f_2	1	0	1	0	

ϕ_1^3

	0	1	2	3	g_3
0	0	1	1	1	1
1	0	1	1	1	1
2	0	1	1	1	1
3	0	1	1	1	1
4	0	1	1	1	1
5	0	1	1	1	1
6	0	1	1	1	1
7	0	1	1	1	1
f_3	0	1	1	1	

ϕ_1

	0	1	2	3
0	0	0	1	0
1	0	1	0	1
2	0	0	1	0
3	0	1	0	1
4	0	0	1	0
5	0	1	0	1
6	0	0	1	0
7	0	1	0	1

Solution matrix for ϕ_1. (Note: if only 1 equation had been given originally, ϕ_1 would represent the $1 \cdot 2 \cdot 1 \cdot 2 \cdot 1 \cdot 2 \cdot 1 \cdot 2 = 8$ different solutions).

ϕ_2^0

	0	1	2	3	g_o
0	1	0	1	0	0
1	0	1	0	1	1
2	1	0	1	0	0
3	0	1	0	1	1
4	1	0	1	0	0
5	0	1	0	1	1
6	1	0	1	0	0
7	0	1	0	1	1
f_o	0	1	0	1	

ϕ_2^1

	0	1	2	3	g_1
0	1	1	1	1	1
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	1
7	1	1	1	1	1
f_1	1	1	1	1	

Figure 10.

ϕ_2^2

	0	1	2	3	g_2
0	1	1	1	0	1
1	0	0	0	1	0
2	1	1	1	0	1
3	1	1	1	0	1
4	1	1	1	0	1
5	0	0	0	1	0
6	1	1	1	0	1
7	1	1	1	0	1
f_2	1	1	1	0	

ϕ_2^3

	0	1	2	3	g_3
0	0	0	1	1	1
1	0	0	1	1	1
2	0	0	1	1	1
3	1	1	0	0	0
4	0	0	1	1	1
5	0	0	1	1	1
6	0	0	1	1	1
7	1	1	0	0	0
f_3	0	0	1	1	

$\underline{\phi_2}$

	0	1	2	3
0	0	0	1	0
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	0	0	1	0
5	0	0	0	1
6	0	0	1	0
7	0	1	0	0

<u>Figure 10.</u>

$$= x_4(\bar{x}_1x_5+x_3\bar{x}_5)+x_4\bar{x}_5(x_2+\bar{x}_3)+x_5(\bar{x}_1\bar{x}_4+\bar{x}_2x_4+\bar{x}_3x_4)$$

$$\equiv g_2(x_1,x_2,\ldots,x_5) \tag{3.12.2}$$

Steps 1 and 2: The maps of ϕ_k are formed in Figure 10. Note that this is a given system of two simultaneous equations, but since $r-m = 5-3 = 2$, there will also be $2^2 = 4$ simultaneous equations per given equation. (Actually, Maitra treats only this latter set as his simultaneous system, and works with only one given equation.)

Step 3: Form ϕ (Figure 11).

	ϕ			
	y_1y_2			
$x_1x_2x_3$	00	01	10	11
0 0 0	0	0	1	0
0 0 1	0	0	0	1
0 1 0	0	0	1	0
0 1 1	0	1	0	0
1 0 0	0	0	1	0
1 0 1	0	0	0	1
1 1 0	0	0	1	0
1 1 1	0	1	0	0

Figure 11.

Step 4: $[\phi] = 1\cdot1\cdot1\cdot1\cdot1\cdot1\cdot1\cdot1 = 1 = T$; therefore a unique solution exists.

Treating Figure 11 as a truth table, the unique answer is immediately seen to be

$$y_1 = \bar{x}_2+\bar{x}_3$$

$$y_2 = x_3$$

Check: A sample check will be shown for $x_4 = x_5 = 1$:

$$f_1 = (\bar{x}_2+\bar{x}_3)x_3\bar{x}_4x_5+(x_2x_3)x_3(x_4x_5+\bar{x}_4\bar{x}_5)+x_4\bar{x}_3(x_5\bar{x}_2+x_5\bar{x}_3+\bar{x}_5x_2x_3)$$

$$+ x_4(\bar{x}_2+\bar{x}_3)(x_5x_3+\bar{x}_5\bar{x}_3)$$

$$= x_2x_3+\bar{x}_2\bar{x}_3+\bar{x}_3+\bar{x}_2x_3 = 1$$

$$g_1 = x_3+x_1\bar{x}_2\bar{x}_3+\bar{x}_1x_2\bar{x}_3+x_1x_2\bar{x}_3+\bar{x}_1\bar{x}_2\bar{x}_3 = 1 \checkmark$$

$$f_2 = (\bar{x}_2+\bar{x}_3)x_3+x_2x_3(x_3)(0)+\bar{x}_3(\bar{x}_2+\bar{x}_3) = \bar{x}_2x_3+\bar{x}_3 = \bar{x}_2+\bar{x}_3$$

$$g_2 = \bar{x}_2 + \bar{x}_3 \checkmark$$

Similar checks for $x_4 = 0$, $x_5 = 1$, $x_4 = 1$, $x_5 = 0$, $x_4 = x_5 = 0$ show that the solution checks identically.

<u>Solution for problems of type 2</u>

Given the functions f and the y_j's, it is desired to determine the function g explicitly. The given conditions imply that the values of f_k and all the elements of the matrices are known. (For further insight, see Ledley's solution of this problem.) Furthermore, since the y_j's are independent of $x_{m+1},\ldots,x_r$, the matrices corresponding to different pairs of f_k's and g_k's (i.e., $\phi^1,\phi^2,\phi^3,\phi^4$ in Example) will be identical (for given y_j's). Hence, to find the values for g_k, it is only necessary to multiply the rows of matrix elements by the corresponding elements of f_k

according to (3.11). Note that if the value of g obtained for each row is not unique, no solution of the original equation exists. This problem may also be solved very easily by substitution of the y_j's into f, and expanding by the usual logical operations. In this way, it may be seen that a solution always exists for this type problem.

Solution for problems of type 3

Given the functions g and y_j's, it is desired to determine the function f explicitly. Once again, the given conditions imply that the values of g_k and the elements of the single ϕ matrix are known. Hence, f_k is found by the multiplications of columns of matrix elements by the corresponding elements of g_k. Also, if the value of f_k obtained for each column is not unique, no solution exists for the original equation. However, a solution with a constraint may be found, but Maitra makes no mention of this.

3.3.3 Ledley's Methods

Robert Ledley's methods incorporate many "new" techniques, such as the use of "designation numbers" and solution by matrix computation. However, when closely examined, it may be seen that he is also using the same ideas as those in the previous sections. Also, the three problem types mentioned in Section 3.3.2 will be

solved, and finally non-matrix solutions for general equations will be shown. Before proceeding, we must examine Ledley's notation.

Terminology: All of the methods are based upon Ledley's digitalization of Boolean algebra. Every possible Boolean function (of the given r variables) is assigned a unique binary number of 2^r bits; this number is called the "designation number" and is written "#". The designation numbers are first assigned to the r variables, or elements. Such an assignment is called a "basis", and is written "b[ ]". A basis for a 3-element system is

$$\begin{array}{llll} & \underline{0123} & \underline{4567} & \\ \#x_1 = & 0101 & 0101 & \\ \#x_2 = & 0011 & 0011 & \\ \#x_3 = & 0000 & 1111 & \end{array}\Bigg\} = b[x_1,x_2,x_3] \qquad (3.13)$$

where the decimal numbers designate the column position. Thus, the basis is just a list of all possible binary combinations of the elements. The basis of (3.13) is written $b[x_1,x_2,x_3]$ and is called a "standard" basis, and it will be assumed that all designation numbers refer to standard bases, unless otherwise indicated. Note that the elements are in "natural" order. All standard bases may be easily formed by alternating 0's and 1's in the first row, pairs

of 0's and 1's in the second, four 0's and four 1's in the third, etc. Observe that a standard basis is nothing more than the BCD code rotated 90° counterclockwise. The significance of the bits is more obscure in a designation number than in a Veitch chart, but the former permits greater ease in many computations. Thus, in a standard basis, the column number in (3.13) corresponds to the binary representation of the column, when read from bottom to top. Finally, subscripts distinguish between the elements (e.g., x_1, x_2), and superscripts denote the column of the bit in a designation number (e.g., $x_3^3 = 0$, $x_1^5 = 1$).

Computations with designation numbers

In order to find the designation number of a given function, it is only necessary to carry out the logical operation on the elements, as indicated by the function. For example, suppose it is desired to find $x_1 + \bar{x}_2$ and $\bar{x}_1 x_3$ (where complementation is simply the replacing of 1's by 0's, and vice versa). From (3.13),

$$
\begin{array}{rlr}
\#x_1 & = 0101 & 0101 \\
\#\bar{x}_2 & = 1100 & 1100 \\
\hline
\#(x_1+\bar{x}_2) & = 1101 & 1101
\end{array}
\qquad\qquad
\begin{array}{rlr}
\#\bar{x}_1 & = 1010 & 1010 \\
\#x_3 & = 0000 & 1111 \\
\hline
\#(\bar{x}_1 x_3) & = 0000 & 1010
\end{array}
$$

It is most important to note that each designation number corresponds to a unique function (simplification not withstanding). Expressed in another way, this says

$$\#x = \#y \quad \text{iff } x=y \tag{3.14}$$

Another important rule is $\#I = 1111....1$ (number of bits depends on the basis), where I is the function whose value is always 1.

Boolean matrix multiplication is denoted by "$\otimes$", and is equivalent to regular matrix multiplication except logical addition and multiplication are used.

The last step in all the methods to be developed is changing back from the solution designation number to the Boolean function of the solution. This may be done in any of the following ways:

1. <u>Disjunctive normal form</u> (first canonical form) - This is simply a sum of complete products. For a basis of r elements, there exist 2^r different complete products. Furthermore, the designation number of a complete product always has a single 1. For the basis (3.13), we have

$$\#(x_1 x_2 x_3) = 0000 \quad 0001$$
$$\#(\bar{x}_1 x_2 x_3) = 0000 \quad 0010$$
$$\#(x_1 \bar{x}_2 x_3) = 0000 \quad 0100$$
$$\#(\bar{x}_1 \bar{x}_2 x_3) = 0000 \quad 1000$$
$$\#(x_1 x_2 \bar{x}_3) = 0001 \quad 0000$$
$$\#(\bar{x}_1 x_2 \bar{x}_3) = 0010 \quad 0000$$
$$\#(x_1 \bar{x}_2 \bar{x}_3) = 0100 \quad 0000$$
$$\#(\bar{x}_1 \bar{x}_2 \bar{x}_3) = 1000 \quad 0000$$

Thus, this form is the sum of the complete products which contribute 1's to the designation number.

2. Conjunctive normal form (second cononical form) - This is a product of complete sums. There are 2^r different complete sums, each one containing only a single zero (the reader should verify this). The Boolean function is thus formed by taking the product of those complete sums which contribute a zero to the designation number.

3. Veitch chart - A Veitch chart may be used quite successfully for the simplification of a Boolean function from its designation number. For example, a 16-bit designation number and its corresponding 4-element standard basis are:

x_1 =	0101	0101	0101	0101
x_2 =	0011	0011	0011	0011
x_3 =	0000	1111	0000	1111
x_4 =	0000	0000	1111	1111
$\#y$ =	0111	0111	1100	1100

The corresponding Veitch chart (see explanation below) is Figure 12.

x_3x_4 \ x_1x_2	00	10	01	11
00	0	1	1	1
10	0	1	1	1
01	1	1	0	0
11	1	1	0	0

$$y = x_2\bar{x}_4 + \bar{x}_2x_4 + x_1\bar{x}_2$$

Figure 12.

By comparing b $[x_1, x_2, x_3, x_4]$ and $\#y$ with Figure 1 and Figure 10, it may be seen that the cells of the chart correspond to the successive groups of four binary combinations of the basis. Furthermore, the cells of the chart may be filled by "folding" the designation number, whence each group of four bits becomes a row. The superiority of a Karnaugh map to a Veitch chart for ordinary simplification is lost here, since the former cannot be formed directly from the

designation number, as can the latter. Thus, the use of a Veitch chart will occur frequently here.

4. Other methods - Ledley also demonstrates various methods which are more artificial and complex than those above. Among these are the generation of "mongrel forms", the use of prime implicants, and "elimination pairs" to determine "included" and "non-included" elements.

Formation of matrices

Returning to the three problem types of Section 3.3.2 we see that it will be necessary to form matrices corresponding to the given functions, and then calculate the solution. The method of solution is summarized below.

1. List the given Boolean functions.
2. Find their designation numbers.
3. Form corresponding Boolean matrices.
4. Compute the Boolean solution matrix with aid of the fundamental formulas.
5. Find solution designation number.
6. Derive explicit Boolean function of solution.

The three matrices of interest are $[g_{km}]$, $[f_{k\ell}]$, and $[\phi_{\ell m}]$, which are defined as follows:

$[g_{km}]$: (a) Form $\#g$ from $b[x_1,x_2,\ldots,x_m,x_{m+1},\ldots,x_r]$. (This applies to problem types 1 and 3.)

(b) Separate the positions of $\#g$ into 2^{r-m} groups of 2^m bits per group.

(c) Index the groups by k ($k = 0,1,\ldots,2^{r-m}-1$). These form the rows of $[g_{km}]$.

(d) Index the bits or positions by m ($m = 0,1,\ldots,2^m-1$). These form the columns of $[g_{km}]$.

Note that these steps simply convert the designation number to a Veitch chart, as was shown above. For that example, the indexing would be

m	0123	0123	0123	0123
#g	= 0111	0111	1100	1100
k	0	1	2	3

$[f_{k\ell}]$: The method is the same as for $[g_{km}]$, except $b[y_1,y_2,\ldots,y_\ell, x_{m+1},\ldots,x_r]$ is used. (This applies to problem types 1 and 2.)

$[O_{\ell m}]$: This is for use in problem types 2 and 3.

(a) Form the $\#y$'s from $b[x_1,x_2,\ldots,x_m]$, and place in successive rows, in numerical order (i.e., $\#y_1,\#y_2,\ldots$).

(b) Index the columns of b $[x_1, x_2, \ldots, x_m]$ by $m(m=0,1,\ldots,2^m-1)$.

(c) The ℓ index is written below each column of the array of #y's, where ℓ is the decimal number equivalent to the binary number formed by the column, when read from bottom to top.

(d) An element $\phi_{\ell m}$ of $[\phi_{\ell m}]$ is a one for all corresponding pairs of the ℓ and m indeces found above. So, if column 3 of the basis corresponds to $\ell = 1$ in the #y's array, the element $\phi_{1,3}$ will be a unit, etc. Thus, there will be the 2^m units in $[\phi_{\ell m}]$, one in each column, all other elements being zero.

<u>Derivation of Boolean function from matrices</u>

#g: This is found by reversing the procedure for finding the $[g_{km}]$ matrix. Simply unfold the matrix to obtain the designation number, and then use one of the aforementioned procedures.

#f: Reverse the procedure for finding the $[f_{k\ell}]$ matrix.

#y's: In this case, multiple sets of solutions are possible. A result array consisting of 2^m columns indexed by $m(m = 0,1,\ldots,2^m-1)$ and ℓ rows indexed by $y_1, y_2, \ldots, y_\ell$ is formed. (The y_j's may be found immediately by treating $[\phi_{\ell m}]$ as a Veitch chart.) For the pairs of

indices ℓ, m for which $\phi_{\ell m} = 1$, place the ℓth column of b $[y_1, y_2, \ldots, y_\ell]$ in the mth column of the result array. The rows of the result array will consist of the $\#y_\ell$'s, with respect to b $[x_1, x_2, \ldots, x_m]$, and the explicit Boolean functions may be obtained.

The methods of this and the previous section will be demonstrated in subsequent examples.

Antecedence and Consequence Solutions

In logical problems, there are two types of solutions to a given equation; these are antecedence and consequence solutions. An antecedence equation implies the given equation; in other words, the truth of an antecedence solution is sufficient for the truth of the given equation. Moreover, an antecedence solution will be false for at least all input conditions for which the given equation will be false, but it is not necessarily equivalent to the given equation.

Consequence solutions can be deduced from (i.e., are implied by) the given equation; hence, the truth of a consequence solution is necessary for the truth of the given equation. Also, a consequence solution will be true for at least all input conditions for which the given equation is true, but it is not necessarily an equivalent equation.

If f_a is an antecedence solution, f_c a consequence solution, and g the given function (f_a, f_c, g are functions, as before), the above conditions are expressed as

$$\text{antecedence:} \quad f_a \longrightarrow g$$

$$\text{consequence:} \quad f \longrightarrow f_c$$

If $f_a \longrightarrow g \longrightarrow f_c$, and $f_a = f_c = f$, then $f = g$. Thus, any solution which is both an antecedence and consequence solution will be the solution of the given equation.

The Fundamental Formulas

Ledley presents three pairs of Boolean matrix equations, an antecedence and consequence formula for each of the three problem types. (For detailed derivation of these formulas, see [10], pp. 428-443.) In the formulas below, the subscripts a and c denote antecedence or consequence solutions, respectively.

Table 3.1

Problem Type	Antecedence Solution	Consequence Solution
1	$[f_{\ell k}] \otimes [\bar{g}_{km}] = [\phi_{\ell m}]$ a	$[\bar{f}_{\ell k}] \otimes [g_{km}] = [\bar{\phi}_{\ell m}]$ c
2	$[f_{k\ell}] \otimes [\phi_{\ell m}] = [g_{km}]$	$[\bar{f}_{k\ell}] \otimes [\phi_{\ell m}] = [\bar{g}_{km}]$
3	$[\phi_{\ell m}] \otimes [\bar{g}_{mk}] = [\bar{f}_{\ell k}]$ a	$[\phi_{\ell m}] \otimes [g_{mk}] = [f_{\ell k}]$ c

(3.15)

Properties of the Solutions

Type 1 problems: The solution will be a set of functions. It is possible that many sets of functions will exist; however, it is also possible that none may exist. In the latter case, a constraint may be found among the independent variables so that a solution does exist (as in Svoboda's algorithm). The constraint may be found as a Boolean row vector from

$$[\ I_\ell\ ] \otimes [\ \phi_{\ell m}\ ] = [\ C_m\ ]$$

where I_ℓ is a row vector of all units. Thus, $[c_m] = \#c$, referred to b $[x_1,x_2,\ldots,x_m]$. New bases constrained by c (where the columns having zeros in #c are disallowed) must now replace b $[x_1,x_2,\ldots,x_r]$ and b $[x_1,x_2,\ldots,x_m]$, and the solution is found as previously described, except m = 0,1,...u-1, where u is the number of units in $[c_m]$.

Type 2 problems: The antecedence and consequence solutions for g are the same, so that either equation may be used. For this type only, we always have f = g.

Type 3 problems: There is always at least one antecedence or consequence solution, and it is in the form of a single function.

If there are N solutions of either type, there will be one which is termed "closest" to g. If such a solution is represented by f_m, then

$$f_n \rightarrow f_m \rightarrow g \qquad \text{(antecedence solutions, } n=1,2,\ldots,N)$$

$$g \rightarrow f_m \rightarrow f_n \qquad \text{(consequence solutions, } n=1,2,\ldots,N).$$

Constraints may also be found between the given functions. The constraint will be found as both an antecedence and consequence solution; if none exists, the functions are independent. It is found by substituting the single column vector I_m for $[g_{mk}]$ in the Type 2 fundamental formulas.

<u>Example of Type 1 Problem</u>

Equations (3.12) will be solved using Ledley's method.

Given: Equations 3.12.

Find: y_1, y_2.

<u>Step 1</u>: Find designation numbers of f_1, f_2, g_1, g_2.

$$f_1 = y_1 y_2 \bar{x}_4 \bar{x}_5 + \bar{y}_1 y_2 (x_4 x_5 + \bar{x}_4 \bar{x}_5) + x_4 \bar{y}_2 (x_5 y_1 + \bar{x}_5 \bar{y}_1) + x_4 y_1 (x_5 y_2 + \bar{x}_5 \bar{y}_2)$$

$$f_2 = y_1 y_2 (\bar{x}_4 + x_5) + \bar{y}_1 y_2 (\bar{x}_4 x_5 + x_4 \bar{x}_5) + \bar{x}_5 \bar{y}_1 (x_4 \bar{y}_2 + \bar{x}_4 y_2) + x_5 \bar{y}_2 (\bar{x}_4 \bar{y}_1 + x_4 y_1)$$

$$+ y_1 \bar{y}_2 (x_4 \bar{x}_5 + \bar{x}_4 x_5)$$

$b\left[y_1, y_2, x_4, x_5\right] =$

$\#y_1 =$	0101	0101	0101	0101
$\#y_2 =$	0011	0011	0011	0011
$\#x_4 =$	0000	1111	0000	1111
$\#x_5 =$	0000	0000	1111	1111
$\#f_1 =$	0011	1100	0000	0111
$\#f_2 =$	0011	1110	1111	0101

$$g_1 = x_3(\bar{x}_4\bar{x}_5+x_4x_5)+\bar{x}_2\bar{x}_3x_4(x_1x_5+\bar{x}_1\bar{x}_5)+x_2\bar{x}_3x_4(\bar{x}_1x_5+x_1\bar{x}_5)$$

$$+\bar{x}_3x_4x_5(x_1x_2+\bar{x}_1\bar{x}_2+\bar{x}_3x_4x_5(\bar{x}_1x_2+x_1\bar{x}_2)$$

$$g_2 = \bar{x}_4(x_1x_5+x_3\bar{x}_5)+x_4\bar{x}_5(x_2+\bar{x}_3)+x_5(\bar{x}_1\bar{x}_4+\bar{x}_2x_4+\bar{x}_3x_4)$$

b $[x_1, x_2, x_3, x_4, x_5]$ =

$\#x_1$ =	0101	0101	0101	0101	0101	0101	0101	0101
$\#x_2$ =	0011	0011	0011	0011	0011	0011	0011	0011
$\#x_3$ =	0000	1111	0000	1111	0000	1111	0000	1111
$\#x_4$ =	0000	0000	1111	1111	0000	0000	1111	1111
$\#x_5$ =	0000	0000	0000	0000	1111	1111	1111	1111
$\#g_1$ =	0000	1111	1111	0000	0000	0000	1111	1111
$\#g_2$ =	0000	1111	1111	0011	1111	1111	1111	1100

Step 2: Find $[f_{\ell k}]$, $[\bar{f}_{\ell k}]$, $[g_{km}]$, $[\bar{g}_{km}]$.

In matrix notation,

$$[A_{jk}] = [A_{kj}]^T$$

where the T designates the transpose of the matrix, and is formed by interchanging rows and columns. Also, complementation

(denoted by $[\bar{A}]$) in a Boolean matrix means replacing 0's with 1's and vice versa. So, we find

$$[f^1_{k\ell}] = \begin{bmatrix} 0011 \\ 1100 \\ 0000 \\ 0111 \end{bmatrix} \qquad [f^1_{\ell k}] = \begin{bmatrix} 0100 \\ 0101 \\ 1001 \\ 1001 \end{bmatrix} \qquad [\bar{f}^1_{\ell k}] = \begin{bmatrix} 1011 \\ 1010 \\ 0110 \\ 0110 \end{bmatrix}$$

$$[f^2_{k\ell}] = \begin{bmatrix} 0011 \\ 1110 \\ 1111 \\ 0101 \end{bmatrix} \qquad [f^2_{\ell k}] = \begin{bmatrix} 0110 \\ 0111 \\ 1110 \\ 1011 \end{bmatrix} \qquad [\bar{f}^2_{\ell k}] = \begin{bmatrix} 1001 \\ 1000 \\ 0001 \\ 0100 \end{bmatrix}$$

(The superscripts denote the corresponding equation number.)

$$[g^1_{km}] = \begin{bmatrix} 0000 & 1111 \\ 1111 & 0000 \\ 0000 & 0000 \\ 1111 & 1111 \end{bmatrix} \qquad [g^2_{km}] = \begin{bmatrix} 0000 & 1111 \\ 1111 & 0011 \\ 1111 & 1111 \\ 1111 & 1100 \end{bmatrix}$$

$$[\bar{g}^1_{km}] = \begin{bmatrix} 1111 & 0000 \\ 0000 & 1111 \\ 1111 & 1111 \\ 0000 & 0000 \end{bmatrix} \qquad [\bar{g}^2_{km}] = \begin{bmatrix} 1111 & 0000 \\ 0000 & 1100 \\ 0000 & 0000 \\ 0000 & 0011 \end{bmatrix}$$

Step 3: Find $[\phi^1]$, $[\phi^2]$, by means of the fundamental formulas:

$$[f_{\ell k}] \otimes [g_{km}] = [\phi_{\ell m}]_a \qquad [f_{\ell k}] \otimes [\bar{g}_{km}] = [\phi_{\ell m}]_c$$

$$\begin{array}{ccccc} [f^1_{\ell k}] & & [\bar{g}^1_{km}] & & \\ \begin{bmatrix} 0100 \\ 0101 \\ 1001 \\ 1001 \end{bmatrix} & \otimes & \begin{bmatrix} 1111 & 0000 \\ 0000 & 1111 \\ 1111 & 1111 \\ 0000 & 0000 \end{bmatrix} & = & \begin{bmatrix} 0000 & 1111 \\ 0000 & 1111 \\ 1111 & 0000 \\ 1111 & 0000 \end{bmatrix} \end{array} = [\Phi^1_{\ell m}]_a$$

$$\begin{array}{ccccc} \bar{f}^1_{\ell k} & & g^1_{km} & & \\ \begin{bmatrix} 1011 \\ 1010 \\ 0110 \\ 0110 \end{bmatrix} & \otimes & \begin{bmatrix} 0000 & 1111 \\ 1111 & 0000 \\ 0000 & 0000 \\ 1111 & 1111 \end{bmatrix} & = & \begin{bmatrix} 1111 & 1111 \\ 0000 & 1111 \\ 1111 & 0000 \\ 1111 & 0000 \end{bmatrix} \end{array} = [\Phi^1_{\ell m}]_c$$

$[\phi^1_{\ell m}] = [\phi^1_{\ell m}]_a \cdot [\phi^1_{\ell m}]_c$

where $\cdot$ is logical multiplication of corresponding elements (units appear only where $[\phi]_a$ and $[\phi]_c$ have 1's).

$$[\phi^1_{\ell m}] = \begin{bmatrix} 0000 & 0000 \\ 1111 & 0000 \\ 0000 & 1111 \\ 0000 & 1111 \end{bmatrix}$$

(Note: If equation 1 were taken alone, there would be $2 \cdot 2 \cdot 2 \cdot 2 = 8$ solutions.)

$$\begin{array}{ccccc} [f^2_{\ell k}] & & [\bar{g}^2_{km}] & & \\ \begin{bmatrix} 0110 \\ 0111 \\ 1110 \\ 1011 \end{bmatrix} & \otimes & \begin{bmatrix} 1111 & 0000 \\ 0000 & 1100 \\ 0000 & 0000 \\ 0000 & 0011 \end{bmatrix} & = & \begin{bmatrix} 0000 & 1100 \\ 0000 & 1111 \\ 1111 & 1100 \\ 1111 & 0011 \end{bmatrix} = [\bar{\phi}^2_{\ell m}]_a \end{array}$$

$$\begin{array}{ccccc} [\bar{f}^2_{\ell k}] & & [g^2_{km}] & & \\ \begin{bmatrix} 1001 \\ 1000 \\ 0001 \\ 0100 \end{bmatrix} & \otimes & \begin{bmatrix} 0000 & 1111 \\ 1111 & 0011 \\ 1111 & 1111 \\ 1111 & 1100 \end{bmatrix} & = & \begin{bmatrix} 1111 & 1111 \\ 0000 & 1111 \\ 1111 & 1100 \\ 1111 & 0011 \end{bmatrix} = [\bar{\phi}^2_{\ell m}]_c \end{array}$$

$$[\phi^2_{\ell m}] = [\phi^2_{\ell m}]_a \cdot [\phi^2_{\ell m}]_c \qquad \text{and}$$

$$\begin{array}{ccccc} [\phi^2_{\ell m}]_a & & [\phi^2_{\ell m}]_c & & [\phi^2_{\ell m}] \\ \begin{bmatrix} 1111 & 0011 \\ 1111 & 0000 \\ 0000 & 0011 \\ 0000 & 1100 \end{bmatrix} & \cdot & \begin{bmatrix} 0000 & 0000 \\ 1111 & 0000 \\ 0000 & 0011 \\ 0000 & 1100 \end{bmatrix} & = & \begin{bmatrix} 0000 & 0000 \\ 1111 & 0000 \\ 0000 & 0011 \\ 0000 & 1100 \end{bmatrix} \end{array}$$

Step 4: Find $[\phi_{\ell m}] = [\phi^1_{\ell m}] \cdot [\phi^2_{\ell m}]$

$$[\phi_{\ell m}] = \begin{bmatrix} 0000 & 0000 \\ 1111 & 0000 \\ 0000 & 0011 \\ 0000 & 1100 \end{bmatrix}$$

(Note there will be a unique solution since there is only 1 unit per column.)

Step 5: Set up solution array and derive Boolean functions for y_1, y_2. (Figure 13).

$$b\ [y_1, y_2] : \quad \begin{array}{rcl} \ell & = & 0\ 1\ 2\ 3 \\ \#y_1 & = & 0\ 1\ 0\ 1 \\ \#y_2 & = & 0\ 0\ 1\ 1 \end{array}$$

(Place the ℓth column of b $[y_1, y_2]$ in the mth column for indices of units in $[\phi_{\ell m}]$.)

Solution Array

m	0	1	2	3	4	5	6	7
$\#y_1$	1	1	1	1	1	1	0	0
$\#y_2$	0	0	0	0	1	1	1	1

Figure 13.

Since $y_{1,2} = y_{1,2}(x_1, x_2, x_3)$, the explicit Boolean functions for y_1 and y_2 are found with a three variable Veitch chart. (Figure 14).

x_3 \ x_1x_2	00	10	01	11
0	1	1	1	1
1	1	1	0	0

$y_1 = \bar{x}_2 + \bar{x}_3$

x_3 \ x_1x_2	00	10	01	11
0	0	0	0	0
1	1	1	1	1

$y_2 = x_3$

Figure 14.

The solutions check with those found by Maitra's methods.

Remarks on Type 1 Solutions

It has been shown that

$$[\phi_{\ell m}] = [\phi_{\ell m}]_a \cdot [\phi_{\ell m}]_c \qquad (3.17)$$

where

$$[\phi_{\ell m}]_a = \overline{[f_{k\ell}]^T \otimes [\bar{g}_{km}]} \qquad (3.18)$$

and

$$[\phi_{\ell m}]_c = \overline{[\bar{f}_{k\ell}]^T \otimes [g_{km}]} \qquad (3.19)$$

Substitution of (3.18) and (3.19) into 3.17 yields

$$\phi_{\ell m} = \overline{\sum_k (f_{\ell k} \cdot \bar{g}_{km})} \cdot \overline{\sum_k (\bar{f}_{\ell k} \cdot g_{km})} \tag{3.20}$$

By De Morgan's rules, (3.20) is

$$\phi_{\ell m} = \prod_k (\bar{f}_{\ell k} + g_{km}) \cdot \prod_k (k_{\ell k} + \bar{g}_{km}) \tag{3.21}$$

$$= \prod_k (\bar{f}_{\ell k} + g_{km}) \cdot (f_{\ell k} + \bar{g}_{km})$$

$$= \prod_k (f_{\ell k} g_{km} + \bar{f}_{\ell k} \bar{g}_{km}) \tag{3.22}$$

Observe that (3.22) is identical to (3.2). Ledley calls this matrix-multiplication operation the "theta product", and writes it

$$[\phi_{\ell m}] = [f_{k\ell}]^T \ominus [g_{km}] \tag{3.23}$$

Using (3.22) in the above example would have immediately yielded

$$\begin{array}{ccccc} [f^1_{\ell k}] & & [g^1_{km}] & & \\ \begin{bmatrix} 0100 \\ 0101 \\ 1001 \\ 1001 \end{bmatrix} & \theta & \begin{bmatrix} 0000 & 1111 \\ 1111 & 0000 \\ 0000 & 0000 \\ 1111 & 1111 \end{bmatrix} & = & \begin{bmatrix} 0000 & 0000 \\ 1111 & 0000 \\ 0000 & 1111 \\ 0000 & 1111 \end{bmatrix} = [\phi^1_{\ell m}] \end{array}$$

(Units appear in $[\phi^1_{\ell m}]$ only where $f^1_{\ell k} = g^1_{km}$, i.e., wherever the row of $f_{\ell k}$ is identical to the column of g^1_{km} it is multiplying.)

Also, by treating $[\phi_{\ell m}]$ as a Veitch chart, the functions may be found immediately. See Figure 15 below.

x_3 x_2 x_1 $y_2 y_1$

0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1
0	0	0	0	1	1	0	0

Figure 15.

$$y_1 = \bar{x}_2 + \bar{x}_3$$

$$y_2 = x_3$$

These improvements (i.e., the use of (3.22) and the Veitch chart) result in the methods of Svoboda and Maitra, both of whom reference Ledley.

Example of Type 2 Problem

Given: $f_1(y_1,y_2,x_4,x_5)$, $f_2(y_1,y_2,x_4,x_5)$ (Equation 3.12), $y_1=\bar{x}_2+\bar{x}_3, y_2=x_3$

Problem: Find $g_1(x_2,x_3,x_4,x_5)$ (Note: since none of the given functions contains x_1, this variable will not appear in g_1,g_2.)

Step 1: Find $[f^1_{k\ell}]$, $[f^2_{k\ell}]$, $[\phi^1_{\ell m}]$, $[\phi^2_{\ell m}]$.

From the previous example,

$$[f^1_{k\ell}] = \begin{bmatrix} 0&0&1&1\\ 1&1&0&0\\ 0&0&0&0\\ 0&1&1&1 \end{bmatrix} \qquad [f^2_{k\ell}] = \begin{bmatrix} 0&0&1&1\\ 1&1&1&0\\ 1&1&1&1\\ 0&1&0&1 \end{bmatrix}$$

$$\begin{array}{rcl} & & \underline{0\ 1\ 2\ 3} \\ \#x_2 &=& 0\ 1\ 0\ 1 \\ \#x_3 &=& 0\ 0\ 1\ 1 \\ \hline \#y_1 &=& 1\ 1\ 1\ 0 \\ \#y_2 &=& \underline{0\ 0\ 1\ 1} \\ & & 1\ 1\ 3\ 2 \end{array}$$

$$\text{Thus } [\phi^1_{\ell m}] \quad [\phi^2_{\ell m}] = \begin{bmatrix} 0&0&0&0\\ 1&1&0&0\\ 0&0&0&1\\ 0&0&1&0 \end{bmatrix} = [\phi_{\ell m}]$$

(Units appear in $[\phi_{\ell m}]$ at (1,0), (1,1), (3,2), (2,3).)

Step 2: Calculate $[g^1_{km}]$ and $[g^2_{km}]$ from the fundamental formula.

$$[f^1_{k\ell}] \otimes [\phi_{\ell m}] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} = [g^1_{km}]$$

$$[f^2_{k\ell}] \otimes [\phi_{\ell m}] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = [g^2_{km}]$$

Step 3: Find $g_1(x_2,x_3,x_4,x_5)$, $g_2(x_2,x_3,x_4,x_5)$ by means of Veitch charts (Figure 14).

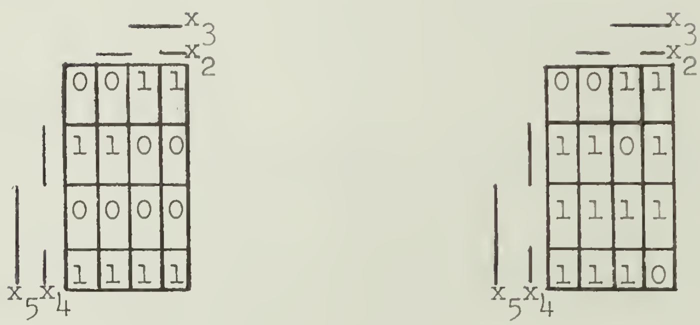


Figure 16.

$$g_1 = x_4x_5+\bar{x}_3x_4+x_3\bar{x}_4\bar{x}_5 \qquad g_2 = \bar{x}_4x_5+\bar{x}_3x_4+x_3\bar{x}_4+x_2x_4\bar{x}_5+\bar{x}_2x_5$$

(Note: These functions may be obtained by (3.12.1) and (3.12.2), respectively, if minimized by De Morgan's rules.)

Example of Type 3 Problem

Given: g_1, g_2 (as in previous problems - Equation 3.12); $y_1 = \bar{x}_2 + \bar{x}_3$

$y_2 = x_3$

Problem: Find $f_1(y_1,y_2,x_4,x_5)$, $f_2(y_1,y_2,x_4,x_5)$.

Step 1: Find $[\bar{g}^1_{mk}]$, $[\bar{g}^2_{mk}]$, $[\phi^1{}_{\ell m}]$, $[\phi^2{}_{\ell m}]$.

From the previous problem,

$$[g^1_{km}] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad [\bar{g}^1_{mk}] = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$[\phi_{\ell m}] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$[g^2_{km}] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad [\bar{g}^2_{mk}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<u>Step 2</u>: Calculate $[\bar{f}^1_{\ell k}]$, $[\bar{f}^2_{\ell k}]$ from the fundamental formula, $[\phi_{\ell m}] \otimes [\bar{g}_{mk}] = [\bar{f}_{\ell k}]_a$. Therefore,

$$[\bar{f}^1_{\ell k}]_a = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad [\bar{f}^2_{\ell k}]_a = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad [f^1_{k\ell}]_a = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad [f^2_{k\ell}]_a = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

If f_1 and f_2 were now found with respect to $b[y_1,y_2,x_4,x_5]$, this would imply that no constraint exists between the elements of the basis, i.e., that they are independent. The constraint with respect to the basis is found by

$$[\phi_{\ell m}] \otimes [I_{mk}] = [c_{\ell k}].$$

$$\begin{bmatrix} 0&0&0&0 \\ 1&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{bmatrix} \otimes \begin{bmatrix} 1&1&1&1 \\ 1&1&1&1 \\ 1&1&1&1 \\ 1&1&1&1 \end{bmatrix} = \begin{bmatrix} 0&0&0&0 \\ 1&1&1&1 \\ 1&1&1&1 \\ 1&1&1&1 \end{bmatrix} = [c_{\ell k}][c_{k\ell}] - \begin{bmatrix} 0&1&1&1 \\ 0&1&1&1 \\ 0&1&1&1 \\ 0&1&1&1 \end{bmatrix}$$

If $[c_{k\ell}]$ is treated as a Veitch chart, the constraint is found immediately as $c = y_1 + y_2 = 1$. This checks, since $\bar{x}_2 + \bar{x}_3 + x_3 = 1$. The f's may now be found from the constrained basis or constrained Veitch chart (binary combinations where #c is zero are disallowed - Figure 17).

$b_c[y_1, y_2, x_4, x_5] =$

$\#y_1$ =	101	101	101	101
$\#y_2$ =	011	011	011	011
$\#x_4$ =	000	111	000	111
$\#x_5$ =	000	000	111	111
$\#f^1_a$ =	011	100	000	111
$\#f^2_a$ =	011	110	111	101

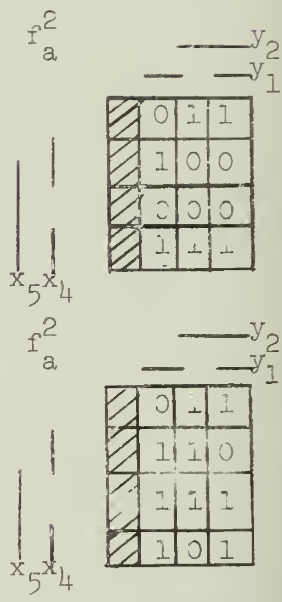


Figure 17.

$$f_a^1 = y_2x_4x_5+y_1\bar{y}_2x_4+y_2\bar{x}_4\bar{x}_5$$

$$f_a^2 = y_2\bar{x}_4+\bar{y}_1y_2\bar{x}_5+y_1x_5+y_1\bar{y}_2x_4$$

At this point, we have $f_a^1 \twoheadrightarrow g_1$, $f_a^2 \twoheadrightarrow g_2$. It is now necessary to use type 2 formulas to see if $\#f_a = \#g$. Thus

$$[f^1_{k\ell}] \otimes [\phi_{\ell m}] = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} = [f^1_{km}] \equiv [g^1_{km}]$$

$$[f^2_{k\ell}] \otimes [\phi_{\ell m}] = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = [f^2_{km}] \equiv [g^2_{km}]$$

The above computation may be taken as the substitution of $y_1(x_2,x_3)$ and $y_2(x_2,x_3)$ into f_a^1 and f_a^2, or the means of displaying that $f_s = g_s$ by showing $f_a^S \twoheadrightarrow g_s \twoheadrightarrow f^S$. Thus, the functions for f_a^1 and f_a^2 are the desired solution (which includes the constraint).

Ledley's Solution of General Functional Equations

Ledley's method is the same as Svoboda's, but through the use of charts for the stepwise calculation of the functions f_k and g_k, followed by matrix multiplication to find ϕ, Ledley's method in comparison seems mysterious, and is more laborious. The explanation of the method will be concurrent with the solution of Equations (3.5).

Step 1: Set up a chart which lists:

(a) the designation numbers of the coefficient of each unknown;

(b) the corresponding unknown;

(c) the value of the unknown computed for all possible binary combinations.

See Table 3.2 on following page.

Step 2: Using Table 3.2, multiply the transpose of each matrix formed in column C by the corresponding matrix formed in column A. This will yield the Boolean matrices $[f_1]$, $[g_1]$, $[f_2]$, $[g_2]$ which are equivalent to the Veitch charts of these functions found in Svoboda's algorithm (see Figure 2).

Table 3.2

	A. Coefficients and Designation Numbers		B. Corresponding Unknowns	C. Binary Combinations of Solution y_1y_2			
				00	10	01	11
f_1	$\#x_1\bar{x}_2 =$	0 1 0 0	y_2	0	0	1	1
	$\#\bar{x}_2 =$	1 1 0 0	y_1y_2	0	0	0	1
	$\#x_2 =$	0 0 1 1	$y_1\bar{y}_2$	0	1	0	0
	$\#x_2 =$	0 0 1 1	$\bar{y}_1y_2$	0	0	1	0
g_1	$\#\bar{x}_1x_2 =$	0 0 1 0	y_2	0	0	1	1
	$\#(\bar{x}_1+x_2) =$	1 0 1 1	$y_1\bar{y}_2$	0	1	0	0
	$\#x_1\bar{x}_2 =$	0 1 0 0	$\bar{y}_1$	1	0	1	0
	$\#x_1 =$	0 1 0 1	$\bar{y}_1y_2$	0	0	1	0
f_2	$\#(x_1\bar{x}_2+\bar{x}_1x_2) =$	0 1 1 0	y_1	0	1	0	1
	$\#I =$	1 1 1 1	$y_1\bar{y}_2$	0	1	0	0
	$\#\bar{x}_1\bar{x}_2 =$	1 0 0 0	$\bar{y}_1$	1	0	1	0
	$\#\bar{x}_2 =$	1 1 0 0	$\bar{y}_2$	1	1	0	0
	$\#x_1x_2 =$	0 0 0 1	$\bar{y}_1y_2$	0	0	1	0
g_2	$\#\bar{x}_2 =$	1 1 0 0	$\bar{y}_2$	1	1	0	0
	$\#I =$	1 1 1 1	$\bar{y}_1\bar{y}_2$	1	0	0	0
	$\#x_1x_2 =$	0 0 0 1	$\bar{y}_1$	1	0	0	1
	$\#x_1\bar{x}_2 =$	0 1 0 0	y_1	0	1	0	1
	$\#\bar{x}_1x_2 =$	0 0 1 0	y_1y_2	0	0	0	1

$$\begin{bmatrix} 0&0&0&0 \\ 0&0&1&0 \\ 1&0&0&1 \\ 1&1&0&0 \end{bmatrix} \otimes \begin{bmatrix} 0&1&0&0 \\ 1&1&0&0 \\ 0&0&1&1 \\ 0&0&1&1 \end{bmatrix} = \begin{bmatrix} 0&0&0&0 \\ 0&0&1&1 \\ 0&1&1&1 \\ 1&1&0&0 \end{bmatrix} = [f_1]$$

$$\begin{bmatrix} 0&0&1&0 \\ 0&1&0&0 \\ 1&0&1&1 \\ 1&0&0&0 \end{bmatrix} \otimes \begin{bmatrix} 0&0&1&0 \\ 1&0&1&1 \\ 0&1&0&0 \\ 0&1&0&1 \end{bmatrix} = \begin{bmatrix} 0&1&0&0 \\ 1&0&1&1 \\ 0&1&1&1 \\ 0&0&1&0 \end{bmatrix} = [g_1]$$

$$\begin{bmatrix} 0&0&1&1&0 \\ 1&1&0&1&0 \\ 0&0&1&0&1 \\ 1&0&0&0&0 \end{bmatrix} \otimes \begin{bmatrix} 0&1&1&0 \\ 1&1&1&1 \\ 1&0&0&0 \\ 1&1&0&0 \\ 0&0&0&1 \end{bmatrix} = \begin{bmatrix} 1&1&0&0 \\ 1&1&1&1 \\ 1&0&0&1 \\ 0&1&1&0 \end{bmatrix} = [f_2]$$

$$\begin{bmatrix} 1&1&1&0&0 \\ 1&0&0&1&0 \\ 0&0&1&0&0 \\ 0&0&0&1&1 \end{bmatrix} \otimes \begin{bmatrix} 1&1&0&0 \\ 1&1&1&1 \\ 0&0&0&1 \\ 0&1&0&0 \\ 0&0&1&0 \end{bmatrix} = \begin{bmatrix} 1&1&1&1 \\ 1&1&0&0 \\ 0&0&0&1 \\ 0&1&1&0 \end{bmatrix} = [g_2]$$

In Steps 1 and 2, Ledley has presented a systematic procedure for the mapping of f_1, g_1, f_2, g_2, which was carried out mentally in Svoboda's algorithm. This procedure is unduly laborious, and seems to present no factor which is superior to Svoboda's use of Veitch charts.

The remaining process would consist of finding $[\phi_1]$ and $[\phi_2]$ based upon (3.2), then finding $[\phi] = [\phi_1] \cdot [\phi_2]$ (where $\cdot$ is logical multiplication of corresponding elements), and finally derivation of the y_j's by the solution array (or a Veitch chart). Since this is analogous to Svoboda's solution, completing the solution here would be superfluous.

3.3.4 Carvallo's Method

Carvallo's [3] method of solution is almost identical to Ledley's alternate method, in the use of a basis and Ledley's "θ matrix multiplication." However, Carvallo's basis is written as a single column matrix of 2^r complete products, and is expressed as

$$\bar{\bar{x}} \quad \begin{bmatrix} x_1 & x_2 \cdots x_r \\ x_1 & x_2 \cdots \bar{x}_r \\ \vdots & \vdots & \vdots \\ \bar{x}_1 & \bar{x}_2 \cdots x_r \\ \bar{x}_1 & \bar{x}_2 \cdots \bar{x}_r \end{bmatrix} = \begin{bmatrix} 1 & 1 \cdots 1 \\ 1 & 1 \cdots 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 \cdots 1 \\ 0 & 0 \cdots 0 \end{bmatrix} \qquad (3.23)$$

Thus, the 3-element basis equivalent to (3.13) is

$$\bar{\bar{x}} = \begin{array}{c} \begin{array}{ccc} (x_1) & (x_2) & (x_3) \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \qquad (3.13')$$

The solution is found from (3.2) or (3.22), but since the designation numbers in the f and g matrices are in <u>columns</u> instead of rows, (3.23) must now be written as

$$[\phi]_{(s,t)} = [f]_{(s,d)} \; \theta \; [g]^T_{(t,d)} \qquad (3.23')$$

where $s = 2^a$, $t = 2^b$, (a and b = number of elements in respective bases) d = number of #'s in matrix (in Ledley's case, d = 1). Also, if the number of elements in the bases is small enough ($r \leq 3$) so that the matrix multiplication is not unwieldly, the entire system of equations may be solved in one step (i.e., it is not necessary to calculate

$[\phi^1]$, $[\phi^2]$,... and finally $[\phi] = \prod_{i=1}^{n} [\phi^i]$). For, if $[f]_{(s,d)} =$ $[\#f_1, \#f_2, \ldots, \#f_d]$, and $[g]_{(t,d)} = [\#g_1, \#g_2, \ldots, \#g_d]$, then

$$[f]_{(s,d)} \; \theta \; [g]^T_{(t,d)} = [\phi^1]_{(s,t)} \cdot [\phi^2]_{(s,t)} \cdots [\phi^n]_{(s,t)} = [\phi]_{(s,t)}.$$

Carvallo does treat the $[\phi]_{(s,t)}$ as a veitch chart and thus derives his solution function immediately.

<u>Example</u>:

Given:

$$f_1(y_1,y_2,y_3) \equiv y_1\bar{y}_2+\bar{y}_1\bar{y}_3 = x_1x_3+\bar{x}_2\bar{x}_3 \equiv g_1(x_1,x_2,x_3) \quad (3.24.1)$$

$$f_2(y_1,y_2,y_3) \equiv \bar{y}_1(y_2+y_3)+y_1\bar{y}_2\bar{y}_3 = \bar{x}_1x_2+\bar{x}_2(x_1+\bar{x}_3) \equiv$$

$$g_2(x_1,x_2,x_3) \quad (3.24.2)$$

$$f_3(y_1,y_2,y_3) \equiv y_3(y_1+y_2)+\bar{y}_1y_2 = x_1(x_2+x_3)+x_2x_3 \equiv$$

$$g_3(x_1,x_2,x_3) \quad (3.24.3)$$

Find: $y_1(x_1,x_2)$, $y_2(x_1,x_2)$.

Step 1: Find $[f]_{(8,3)}$, $[g]^T_{(3,8)}$ $\quad (s = 2^3 = 8,\ t = 2^3 = 8,\ d = 3)$

$$[f] = \begin{array}{c} \begin{array}{ccc} (\#f_1) & (\#f_2) & (\#f_3) \end{array} \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \end{array} \qquad [g]^T = \begin{array}{cc} \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} & \begin{array}{c} (\#g_1) \\ (\#g_2) \\ (\#g_3) \end{array} \end{array}$$

Step 2: Find $[\phi]_{(8,8)} = [f] \,\theta\, [g]$ (Figure 18).

$$\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array}$$

$$y_1 y_2 y_3 \quad \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = [\phi]$$

Figure 18.

Step 3: Find y_1, y_2.

By treating $[\phi]$ as a Veitch chart, immediately

$$y_1 = x_1x_2+\bar{x}_2(\bar{x}_1+\bar{x}_3)$$

$$y_2 = x_1x_2\bar{x}_3+x_3(\bar{x}_2+\bar{x}_1)$$

$$y_3 = x_2$$

Note that the variables in the above Veitch chart are the complements of the standard notation used throughout this paper. This was necessitated by the fact that Carvallo's basis is the reverse of the BCD code used by Ledley.

3.4. "Truth-Table-Logic" Method

Given the set of relations (3.1), it is also well known that they may be combined into the single equation

$$\bar{\phi} = \sum_{k=1}^{n} (f_k \oplus g_k) = 0 \qquad (3.25)$$

which is simply the complement of (3.2). This equation may then be expressed as a sum of minterms of $x_1, x_2, \ldots, x_r$ multiplied by coefficients which are functions of $y_1, y_2, \ldots, y_s$. This method of

attack is the basis of a laborious algebra (whose applications are extended in [4]) which is very similar to the method developed in Section 3.5. In order to avoid duplication, only the latter will be presented, and attention will be given to Phister's [15] more useful method.

Phister's other technique is that of "truth-table-logic". It simply defines a systematic procedure for deriving a truth table, containing all binary combinations of the independent variables, by logical reasoning. Although this technique is most obvious, it has limitations: the equations must be of the form $f(y_1,y_2,\ldots,y_s, x_1,x_2,\ldots,x_r) = g(x_1,x_2,\ldots,x_r)$; it is only good for simple equations with few unknowns, such as the input equations for memory elements (Phister's examples are only for flip-flops); constraints are necessary for a general solution. The steps are:

Step 1: Record all input combinations of the independent variables and the corresponding truth values of $g(x_1,x_2,\ldots,x_r)$.

Step 2: By logical reasoning, determine truth values (including don't-cares) of the dependent variables.

Step 3: Form the general solution of y_j's by regular truth table technique.

Step 4: By means of a Karnaugh map, find the simplest solution by assigning specific values to the don't-cares.

The simple example below was taken from [15], pp. 121-124, and the truth table below is Steps 1 and 2 combined.

Example: Given the equations of an R-S flip-flop

$$S + \overline{R}Q = g_1 Q + g_2 \overline{Q}$$

$$RS = 0$$

where g_1 and g_2 represent Boolean functions of whatever variables determine the state of Q; g_1, g_2, and Q are the independent variables.

Steps 1 and 2: The derived truth table is Table 3.3 below.

Step 3: Find the general solutions.

From Table 3.3,

$$R = d_0\overline{g}_1\overline{g}_2\overline{Q} + \overline{g}_1Q + d_4g_1\overline{g}_2\overline{Q}$$

$$S = g_2\overline{Q} + d_5g_1\overline{g}_2Q + d_7g_1g_2Q$$

(These may be checked by substitution into the given equations.)

Table 3.3

g_1	g_2	Q	$g_1Q+g_2\bar{Q}$ $= S + \bar{R}Q$	R	S
0	0	0	0	d_0	0
0	0	1	0	1	0
0	1	0	1	0	1
0	1	1	0	1	0
1	0	0	0	d_4	0
1	0	1	1	0	d_5
1	1	0	1	0	1
1	1	1	1	0	d_7

Step 4: Simplify the functions by means of Karnaugh maps (Figure 19).

			g_2	g_2
	d_0	1	1	0
g_1	d_4	0	0	0
		Q	Q	

$R = \bar{g}_1 Q$

$d_0=d_4=d_5=d_7=0$

			g_2	g_2
	0	0	0	1
g_1	0	d_5	d_7	1
		Q	Q	

$S = g_2 \bar{Q}$

Figure 19

3.5 Parametric Method

S. Rudeanu [18], [19] obtains the general solution of functional equations in parametric form, and also expresses the solution as an explicit function of the given independent variables.

It has been previously shown that a system of equations (3.1) may be expressed as

$$f(y_1,y_2,\ldots,y_s,x_1,x_2,\ldots,x_r) = 0 \tag{3.26}$$

by means of (3.25), (the complement of (3.1)). Furthermore, it is well known (cf. eg., [3], [15], [19]) that a Boolean equation in a single unknown may be written as

$$ay + b\bar{y} = 0 \tag{3.27}$$

for which

$$ab = 0 \tag{3.28}$$

is the necessary and sufficient condition for the existence of a solution. The solutions of (3.27) are constrained by

$$b \leq y \leq \bar{a} \tag{3.29}$$

and all such y's are solutions.

Theorem 3.3: The parametric solution of (3.27) is

$$y = b + p \tag{3.30}$$

where the parameter p is constrained by

$$p \leq \bar{a}\ \bar{b} \tag{3.31}$$

Proof: Every y satisfying (3.30) and (3.31) satisfies (3.29) since $b \leq b + p \leq b + \bar{a}\ \bar{b} = b + \bar{a} = a$ with the aid of the complement of (3.28). Conversely, every y satisfying (3.29) can be written as (3.30) with a suitably chosen p, for if $p = \bar{b}y \leq \bar{a}\ \bar{b}$, then $b + p = b + y = y$. The algorithm for finding the parametric solution is as follows:

Step 1: By means of (3.25), change the system of equations to the form of (3.26) and call this f_1 (all variables present), so

$$f_1(y_1,y_2,\ldots,y_s) = 0 \tag{3.32}$$

where the independent variables are implicitly understood.

Step 2: (3.32) is consistent in y_1 iff

$$f_2(y_2,\ldots,y_s) \equiv f_1(0,y_2,\ldots,y_s)\, f_1(1,y_2,\ldots,y_s) = 0 \quad (3.33)$$

Thus, in the i-th step, find

$$f_{i+1}(y_{i+1},\ldots,y_s) = f_i(0,y_{i+1},\ldots,y_s) f_i(1,y_{i+1},\ldots,y_s). \quad (3.34)$$

Proceed with the successive elimination of dependent variables until finally,

$$f_n(y_n) \equiv f_{n-1}(0,y_n) f_{n-1}(1,y_n) = 0 \quad (3.35)$$

Step 3: Solve (3.35) for y_n by means of (3.30) and (3.31).

Step 4: Substitute the solution for y_n into $f_{n-1}(y_{n-1},y_n) = 0$ to obtain an equation in the single unknown y_{n-1}:

$$f_{n-1}(y_{n-1},p) = 0 \quad (3.36)$$

By successive reintroduction of the unknowns (reversal of Step 2), find $y_1,y_2,\ldots,y_s$ in parametric form.

Step 5: Find the explicit solutions in the form of (3.3) by stepwise tree-like deductions.

Example: Solve the system of equations

$$f_a(y_1,y_2) \equiv y_1+\bar{y}_2x_2 = y_1(\bar{x}_1+\bar{y}_2+x_2)+y_2(\bar{x}_1x_2+\bar{y}_1x_1\bar{x}_2) \equiv g_a(y_1,y_2)$$

$$f_b(y_1,y_2) \equiv \bar{y}_1(y_2+x_2+\bar{x}_1)+x_1x_2 = y_2(\bar{y}_1+\bar{x}_2)+y_1\bar{y}_2(x_1+x_2) \equiv g_b(y_1,y_2)$$

Step 1:

$$f_1(y_1,y_2) = \sum_{i=a}^{b} (f_i \oplus g_i)$$

$$= [y_1+\bar{y}_2x_2][(\bar{y}_1+y_2x_1\bar{x}_2)(\bar{y}_2+y_1x_1+y_1\bar{x}_2+x_1x_2+\bar{x}_1\bar{x}_2)]+[\bar{y}_1(y_2+\bar{x}_2)]$$

$$[y_1(\bar{x}_1+\bar{y}_2+x_2)+y_2(\bar{x}_1x_2+\bar{y}_1x_1\bar{x}_2)]+ [\bar{y}_1(y_2+\bar{x}_1+x_2)+x_1x_2]$$

$$[(\bar{y}_2+y_1x_2)\cdot \bar{x}_1\bar{x}_2(\bar{y}_1+y_2)]+ [(y_1+\bar{y}_2x_1\bar{x}_2)(\bar{x}_1+\bar{x}_2)]\ [y_2(\bar{y}_1+\bar{x}_2)$$

$$+y_1\bar{y}_2(x_1+x_2)]$$

$$= y_1[y_2(x_1+\bar{x}_2)+\bar{y}_2(x_1\bar{x}_2+\bar{x}_1x_2)] +\bar{y}_1[y_2(x_1\bar{x}_2+\bar{x}_1x_2)+\bar{y}_2(\bar{x}_1+x_2)=0$$

Step 2:

$$f_2(y_2) = f_1(0,y_2)f_1(1,y_2)$$

$$= [y_2(x_1\bar{x}_2+\bar{x}_1x_2)+\bar{y}_2(\bar{x}_1+x_2)]\cdot$$

$$[y_2(x_1+\bar{x}_2)+\bar{y}_2(x_1\bar{x}_2+\bar{x}_1x_2)]$$

$$= x_1\bar{x}_2y_2+\bar{x}_1x_2\bar{y}_2$$

Step 3: $a = x_1\bar{x}_2$ $\quad b = \bar{x}_1x_2$ $\quad p \leq (\bar{x}_1+x_2)(x_1+\bar{x}_2) = x_1x_2+\bar{x}_1\bar{x}_2$

Thus,

$$y_2 = b + p = \bar{x}_1x_2 + p$$

$$p = 0 \text{ or } x_1x_2 \text{ or } \bar{x}_1\bar{x}_2 \text{ or } x_1x_2+\bar{x}_1\bar{x}_2$$

and there are four different solutions for y_2.

Step 4: $f_1(y_1,p) = y_1\ [(\bar{x}_1x_2+p)(x_1+\bar{x}_2)+\bar{p}(x_1+\bar{x}_2)(x_1\bar{x}_2+\bar{x}_1x_2)]\ +$

$$\bar{y}_1\ [(\bar{x}_1x_2+p)(x_1\bar{x}_2+\bar{x}_1x_2)+\bar{p}(x_1+\bar{x}_2)(\bar{x}_1+x_2)]$$

$$= [(x_1+\bar{x}_2)p+x_1\bar{x}_2\bar{p}]\,y_1+[\bar{x}_1x_2+x_1\bar{x}_2p+\bar{p}(\bar{x}_1\bar{x}_2+x_1x_2)]\,\bar{y}_1$$

Thus, $$y_1 = \bar{x}_1x_2+x_1\bar{x}_2p+\bar{p}(\bar{x}_1\bar{x}_2+x_1x_2) + q$$

where $$q \leq \bar{a}\,\bar{b} = 0$$

Step 5: Find the explicit functions for y_1 and y_2.

(a) For $p = 0$, $$y_2 = \bar{x}_1x_2$$

$$y_1 = \bar{x}_1 + x_2$$

(b) $p = x_1x_2$, $$y_2 = x_2$$

$$y_1 = \bar{x}_1$$

(c) $p = \bar{x}_1\bar{x}_2$, $$y_2 = \bar{x}_1$$

$$y_1 = x_2$$

(d) $p = x_1x_2+\bar{x}_1\bar{x}_2$, $$y_2 = \bar{x}_1 + x_2$$

$$y_1 = \bar{x}_1x_2$$

Remarks:

1. The parametric form may be avoided by immediately solving for all possible values of y_n and then substituting these into the successive equations $f_{n-1}(y_{n-1}, y_n)$, $f_{n-2}(y_{n-2}, y_{n-1}, y_n)$, etc. This results in a more discernible solution tree.

2. The labor of Step 1 is equivalent to that of Step 1 in Ashenhurst's algorithm, but the remainder of the solution is easier in the latter methods. Consequently, the parametric method does not seem to offer any advantages, and its use should be very limited.

3.6 Summary and Applications

Of all the methods of Section 3, Svoboda's algorithm is probably the most efficient, while still maintaining generality. For the specific type of Equation (3.10), Maitra's methods are simpler than those of Ledley, although Ledley's solutions seem less complex as one practices using them, and "θ products" ease the number of calculations, as in Carvallo's method. The techniques of Ashenhurst and Phister, although the most basic, should be employed to solve only the simplest equations with very few unknowns, and the former should be used in preference to Rudeanu's parametric method.

The most obvious and widespread use of the solution methods for simultaneous Boolean functional equations is logical circuit design. The three problem types solved by Maitra and Ledley were explicitly shown as circuit design problems, as was the flip-flop problem solved by Phister. Moreover, the general methods of Svoboda and Ashenhurst can be used to solve any type of logical circuit design problem.

George Boole [2] devised Boolean algebra as a symbolic interpretation of logical reasoning. There are innumerable everyday problems, in such diverse fields as biochemistry, tactical warfare, economics, and psychology, to which Boolean equations may be applied. These are called word, logic, or sentential problems, and are based upon the feasability of expressing a given proposition as a Boolean function. Table 3.4 below, taken from p. 318 of [4], lists the translations of English connectives, which are used to formulate the problem.

Table 3.4

English connective	Logical Translation
Not A....................................	$\bar{A}$
A and B..................................	$A \cdot B$
A or (inclusive) B; A or B or both............	$A + B$

A or (exclusive) B; either A or B..........$A \neq B$, $A \cdot \overline{B} + \overline{A} \cdot B$, $A \oplus B$

A but B..................................$A \cdot B$

A although B.............................$A \cdot B$

A unless B...............................$\overline{B} \rightarrow A$, $B + A$

A on condition that B....................$B \rightarrow A$, $\overline{B} + A$

A if B...................................$B \rightarrow A$, $\overline{B} + A$

A implies B; if A, then B................$A \rightarrow B$, $\overline{A} + B$

A only if B..............................$A \rightarrow B$, $\overline{A} + B$

Not unless A then B......................$\overline{A} \rightarrow \overline{B}$, $A + \overline{B}$

A provided that B........................$B \rightarrow A$, $\overline{B} + A$

A as well as B...........................$A \cdot B$

A if and only if B.......................$A = B$, $A \cdot B + \overline{A} \cdot \overline{B}$

Not both A and B.........................$\overline{A \cdot B}$, $\overline{A} + \overline{B}$

Neither A nor B..........................$\overline{A} \cdot \overline{B}$, $\overline{A + B}$

When A, then B...........................$A \rightarrow B$, $\overline{A} + B$

A because B..............................$B \rightarrow A$, $\overline{B} + A$

Problems of this type may be used to investigate consistency, redundancy, or relationships between facts, or to find simultaneous equivalences and implications. (See Section 5.2). Ledley's antecedence and consequence solutions are especially useful in problems concerning deduction techniques.

Many fine examples may be found in [10], pp. 405-415, 476-483, and in other sources quoted therein. For a basic discussion, see [20].

The use of simultaneous Boolean equations in the design of sequential machines is suggested by the method in a paper by N. Rouche [17]. Briefly, given a system of equations of the form (3.3), (in previous methods, this was the solution), Rouche transforms

$$
\begin{aligned}
x_1 &= x_1(y_1,y_2,\dots,y_s) \\
x_2 &= x_2(y_1,y_2,\dots,y_s) \\
&\vdots \\
x_r &= x_r(y_1,y_2,\dots,y_s)
\end{aligned}
\qquad (3.37)
$$

into a matrix representation of a sum of minterms. The derivation is based on (3.27) - since only the use of the matrix is of importance here, the derivation will not be discussed, but it should be obvious - and results in a matrix which is actually a Veitch chart mapping $x_i = f_i(y_1,y_2,\dots,y_s)$. For example, if $r = s = 2$ in (3.3), then the given (mapping) equations are

$$y_1 = b_{11}\bar{x}_1\bar{x}_2 + b_{12}\bar{x}_1x_2 + b_{13}x_1\bar{x}_2 + b_{14}x_1x_2 \qquad (3.38.1)$$

$$y_2 = b_{21}\bar{x}_1\bar{x}_2 + b_{22}\bar{x}_1 x_2 + b_{23}x_1\bar{x}_2 + b_{24}x_1 x_2 \tag{3.38.2}$$

and the matrix-Veitch-chart is

$$[B] = \begin{array}{c} \\ \\ y_1 y_2 \end{array} \begin{array}{c} \hphantom{x} \\ \begin{bmatrix} \bar{b}_{11}\bar{b}_{21} & \bar{b}_{12}\bar{b}_{22} & \bar{b}_{13}\bar{b}_{23} & \bar{b}_{14}\bar{b}_{24} \\ \bar{b}_{11}b_{21} & \bar{b}_{12}b_{22} & \bar{b}_{13}b_{23} & \bar{b}_{14}b_{24} \\ b_{11}\bar{b}_{21} & b_{12}\bar{b}_{22} & b_{13}\bar{b}_{23} & b_{14}\bar{b}_{24} \\ b_{11}b_{21} & b_{12}b_{22} & b_{13}b_{23} & b_{14}b_{24} \end{bmatrix} \end{array} \begin{array}{c} x_1 \\ x_2 \\ \\ \\ \\ \end{array} \tag{3.39}$$

Ledley points out that [B] is only a special case of his $[R_{ji}]$ matrix (denoted as $[\phi_{ji}]$ in this paper; see [10], pp. 356-360, 396-400) or, in other words, the problem is a special case of the antecedence and consequence problem. It was shown that in Type 2 and Type 3 problems (in which the y_j's are given), $[\phi_{ji}]$ has a single unit per column; this is also true of [B]. Also [B] is equivalent to the transition matrix in

"The Theory of Nets," by F. Hohn, S. Seshu, and D. Aufenkamp, IRE Transactions on Electronic Computers, Volume EC-6, September 1957, pp. 154 - 161.

If the y_j's are interpreted as the next state (i.e., the state of a machine after an advancing pulse t) and the x_i's as the present state (before the same pulse t), then the matrices [B] and $[\phi_{ji}]$ represent the "state matrix" of an autonomous sequential machine, and the state diagram for the machine may be drawn. If the machine is in present state n, the next state will correspond to the row subscript of the single unit in column n.

Example: Find the state diagram for the machine defined by [B] .

$$[B] = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The equations defining this machine are:

$$y_1 = \bar{x}_1 x_2 + x_1 x_3$$

$$y_2 = \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 x_3$$

$$y_3 = x_2 + x_1 \bar{x}_3$$

Since row 7 has no units, it must be an initial state. The state diagram will be found starting with state 7 (figure 20).

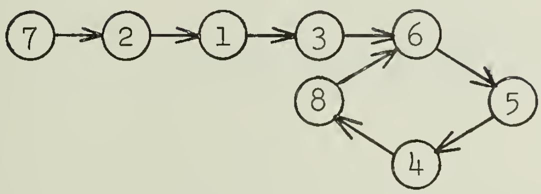


Figure 20.

4. PARTICULAR SOLUTIONS OF GENERAL BOOLEAN EQUATIONS

The methods of this chapter are concerned with solutions which assign a definite value (0 or 1) to each bivalent variable of the simultaneous equations.

4.1 Logical Algebraic Methods

4.1.1. Ashenhurst's Algorithm

This method is simply the special case of the algorithm outlined in section 3.2, and is based only on theorem 3.1. It is only necessary to find ϕ by (3.2) and expand it in canonical form. Then, each term of the expansion will yield a solution, where primed variables are assigned zero values, and unprimed variables have the value unity.

Since the maximum number of solutions is 2^n (for n variables), this method is most efficient when the number of terms in the canonical expansion is much less than 2^n. For the special cases where $g_k = 1$ or 0, the amount of work required to find ϕ may be reduced. First note that if $g_k = 1$, then $f_k g_k + \bar{f}_k \bar{g}_k = f_k$. Secondly, if $g_k = 0$, then $f_k g_k + \bar{f}_k \bar{g}_k = \bar{f}_k$, and in (3.1), $f_k = 0$. Thus, any term in the canonical expansion of ϕ which contains a product appearing in the expansion of f_k as an OR-polynomial must

be zero. Hence, by forming the function ϕ^*, as in (3.2) without using any equations of (3.1) which are of the form $f_k = 0$, ϕ^* yields ϕ if all the products containing a vanishing term are eliminated from the canonical expansion of ϕ^*.

Example: Given the equations

$$x_2+x_6 = \bar{x}_1+\bar{x}_3$$

$$\bar{x}_5(\bar{x}_2+\bar{x}_3) = 0$$

$$x_3\bar{x}_4\bar{x}_6 = 1$$

$$x_1+x_3+x_4\bar{x}_6 = x_5+x_6+\bar{x}_2x_3$$

Step 1: Find ϕ^* (do not use the second equation).

$$\phi^* = [\,(x_2+x_6)(\bar{x}_1+\bar{x}_3)+x_1\bar{x}_2x_3\bar{x}_6\,][\,x_3\bar{x}_4\bar{x}_6\,][\,(x_1+x_3+x_4\bar{x}_6)(x_5+x_6+\bar{x}_2x_3)+$$

$$\bar{x}_1\bar{x}_3(\bar{x}_4+x_6)\bar{x}_5\bar{x}_6(x_2+\bar{x}_3)\,]$$

$$= (\bar{x}_1x_2+\bar{x}_1x_6+x_2\bar{x}_3+\bar{x}_3x_6+x_1\bar{x}_2x_3\bar{x}_6)(x_3\bar{x}_4\bar{x}_6)(x_1x_5+x_1x_6 +$$

$$\bar{x}_2x_3+x_3x_5+x_3x_6+x_4x_5\bar{x}_6+\bar{x}_1\bar{x}_3\bar{x}_4\bar{x}_5\bar{x}_6)$$

$$= \bar{x}_1x_2x_3\bar{x}_4x_5\bar{x}_6+x_1\bar{x}_2x_3\bar{x}_4\bar{x}_6$$

$$\phi^* = \bar{x}_1x_2x_3\bar{x}_4x_5\bar{x}_6 + x_1\bar{x}_2x_3\bar{x}_4\bar{x}_5\bar{x}_6 + x_1\bar{x}_2x_3\bar{x}_4x_5\bar{x}_6$$

Step 2: Eliminate any minterms containing the terms of the second equation. The middle term of ϕ^* will vanish, since it contains $\bar{x}_2\bar{x}_5$.

Step 3: The solutions are:

(a) $x_1 = x_4 = x_6 = 0 \qquad x_2 = x_3 = x_5 = 1$

(b) $x_2 = x_4 = x_6 = 0 \qquad x_1 = x_3 = x_5 = 1$

4.1.2. Grigor'yan's Algorithm

Grigor'yan [6] has proposed the following algorithm for computer solutions of simultaneous Boolean equations. It is based on set theory, but it will be shown that the corresponding logic theory leads directly to Ashenhurst's method.

Let U_i $(i=1,2,\ldots,k)$ be any subsets of some set M, which has a system of characteristic functions

$$Y_i(x) = \begin{cases} 1, \text{ if } x \in U_i \\ 0, \text{ if } x \notin U_i \end{cases} \qquad (x \in M;\ i = 1,2,\ldots,k) \quad (4.1)$$

Suppose we require $Y_i = \sigma_i$, where $\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_k$ is an arbitrary sequence of 0's and 1's; this results in a system of characteristic equations which are satisfied by a given x if and only if

$$\begin{aligned} & x \in U_i \quad \text{whenever} \quad \sigma_i = 1 \\ \text{and} \qquad & x \notin U_i \quad \text{whenever} \quad \sigma_i = 0 \end{aligned} \qquad (4.2)$$

Thus, the given equations $Y_i = \sigma_i$ determine a unique subset of M. Indeed, if (4.2) is true for all $x \in G$, $(G \subset M)$, and for no other x, then G will be the solution. A necessary and sufficient condition for the existence of a solution is obviously that G is not the empty set.

Let q and p be the numbers of 1's and 0's, respectively, in the sequence of σ's. $\cup$, $\cap$, and $\setminus$ are the set-theoretic

operations of union, intersection, and relative complementation (subtraction), respectively. Then G will be found as

$$G = A_q \setminus A_q \cap B_p \tag{4.3}$$

where

$$(o = 1):\ A_q = \bigcap_{\alpha=1} U_{i_\alpha} \quad ; \qquad (o = 0):\ B_p = \bigcup_{\beta=1} U_{i_{\alpha+\beta}} \tag{4.4}$$

Note that depending on the U_i sets, G may be empty, finite, or infinite, resulting in no solutions, a finite number of solutions, or on infinitude of solutions, respectively.

A system of Boolean functions may be uniquely represented in disjunctive normal form, such as

$$Y_i = \sum_{j=0}^{\ell} F_{ij} N_j \qquad (i=1,2,\ldots,k; \quad \ell=2^m-1) \tag{4.5}$$

$$\left(j = \sum_{r=1}^{m} x_r\, 2^{m-r}\right) \tag{4.6}$$

where N_j is the minterm whose m binary variables correspond to the decimal number given by j; each coefficient F_{ij} has a value of 0 or 1 equal to the value of the function Y_i for the value assignment that makes the value of N_j equal to one.

Every logical function Y_i in (4.5) can be put in a one-to-one correspondence with a subset $U_i \subset M$, where $M = \{0,1,2,\ldots,2^m-1\}$, and where U_i is the set of decimal numbers corresponding to the minterms in Y_i. Therefore,

$$Y_i(j) = \begin{cases} 1, & \text{if } j \in U_i \\ 0, & \text{if } j \notin U_i \end{cases} \qquad (4.7)$$

The solution of the set of equations $Y_i = \sigma_i$ $(i=1,2,\ldots,n)$ may now be found from (4.3) and (4.4).

Example: Given a system of 6 equations in 100 variables; they will be represented as in (4.5), but for convenience, only the decimal representation (i.e., j) of the minterm will be used.

$$Y_1 = (0,\ 3,\ 6,\ 10,\ 13,\ 22,\ 36,\ 72,\ 209,\ 504)$$

$$Y_2 = (2,\ 6,\ 9,\ 13,\ 19,\ 22,\ 86,\ 209,\ 411,\ 1031)$$

$$Y_3 = (1,\ 3,\ 6,\ 7,\ 13,\ 22,\ 100,\ 158,\ 209,\ 1000)$$

$$Y_4 = (0,\ 5,\ 6,\ 13,\ 15,\ 19,\ 22,\ 77,\ 198,\ 209,\ 612)$$

$$Y_5 = (5, 6, 13, 28, 94, 106, 601, 2^{100}-1)$$

$$Y_6 = (6, 25, 63, 111, 444, 980, 4000)$$

Case 1: $Y_1 = Y_2 = Y_3 = Y_4 = 1, \quad Y_5 = Y_6 = 0$

Step 1: Find A_q and B_p

$A_q = (6, 13, 22, 209)$

$B_p = (5, 6, 13, 25, 28, 63, 94, 106, 111, 444, 601, 980, 4000, 2^{100}-1)$

Step 2: Find $G = A_q \setminus A_q \cap B_p$

$$G = (6, 13, 22, 209) \setminus (6, 13) = (22, 209)$$

These are the two solutions.

Step 3: Assign binary values to the variables:

In binary, $22 = 0 \cdots 0010110 \therefore x_{96}=x_{98}=x_{99}=1$, all other x's = 0.

$209 = 0 \cdots 011010001 \therefore x_{93}=x_{94}=x_{96}=x_{100}=1$, all other x's = 0.

<u>Case 2</u>: If $Y_1=Y_2=Y_3=Y_4=Y_5=Y_6=1$, $A_q \cap B_p$ is the empty set, and $G = A_q$. In this case, $A_q=6$, which is the unique solution.

Instead of finding (4.3) by set-theoretic operations, the corresponding logical function may be found as

$$g = a_q \cdot (\overline{a_q \cdot \overline{b}_p}) \qquad (4.8)$$

where

$$a_q = \prod_{a=1}^{q} Y_{i_a} \qquad b_p = \sum_{\beta=1}^{p} Y_{i_{q+\beta}} \qquad (4.9)$$

If the given equations are in the form of (3.1), they may be converted to the characteristic functions $Y_i = 1$ by forming

$$Y_i = f_i g_i + \overline{f}_i \overline{g}_i = 1 \qquad (4.10)$$

and thus $b_p=0$. Expanding (4.8) yields in this case

$$g = a_q(\overline{a}_q + \overline{b}_p) = a_q \overline{b}_p = a_q \qquad (4.11)$$

and finally, from (4.9),

$$g = \prod_{a=1}^{g} (f_i g_i + \overline{f}_i \overline{g}_i) \qquad (4.12)$$

which is identical to (3.2). Thus, the method for finding the particular solutions would be that of the preceding section.

4.2. "Map" Method

This method by Nadler,[13] is just a simple case of Svoboda's method. Nadler assumes equations of the form

$$f_i(x_1,x_2,\ldots,x_r) = 0 \qquad (i=1,2,\ldots,n) \qquad (4.13)$$

and maps each function on a Veitch chart (a Karnaugh map may also be used). The solution is then found as the intersection of the n maps. For the more general equations of the form (3.1), the equivalent method would be to form n maps of the Boolean function ϕ_k, as in (3.8); then find the intersection of these maps as ϕ. These are precisely the first two steps in Svoboda's algorithm. Furthermore, Nadler mentions, as did Svoboda, that $\bar{\Phi}$ may be found on a single chart by mapping the complements of the equations (see remark 1, Svoboda's algorithm). For (3.1), this would be done by mapping $f_k \oplus g_k$.

<u>Example</u>: Given the equations

$$f_1(x_1,x_2,x_3,x_4) \equiv x_1\bar{x}_2+(x_1+\bar{x}_2+\bar{x}_3)x_4+\bar{x}_1x_2\bar{x}_3 = x_1x_2+(\bar{x}_2+x_3)\bar{x}_4$$

$$\equiv g_1(x_1,x_2,x_3,x_4)$$

$$f_2(x_1,x_2,x_3,x_4) \equiv x_1\bar{x}_2+\bar{x}_1x_2+\bar{x}_2(\bar{x}_3\bar{x}_4+x_3x_4) = x_2(\bar{x}_1+x_3+\bar{x}_4) +$$

$$x_1(x_3\bar{x}_4+\bar{x}_2\bar{x}_3x_4) \equiv g_2(x_1,x_2,x_3,x_4)$$

<u>Step 1</u>: Map f_1, g_1, f_2, g_2 (fig. 21).

x_2, x_1 (columns); x_4, x_3 (rows)

f_1

0	1	1	0
0	1	0	0
1	1	1	1
1	1	0	1

g_1

1	1	0	1
1	1	1	1
0	0	0	1
0	0	0	1

f_2

1	1	1	0
0	1	1	0
0	1	1	0
1	1	1	0

g_2

0	0	1	1
0	1	1	1
0	1	1	0
0	0	1	1

<u>Fig. 21.</u>

Step 2: Finding $\phi = \phi_1 \cdot \phi_2$ was shown in Svoboda's example, so this example will demonstrate finding $\bar{\phi} = \bar{\phi}_1 + \bar{\phi}_2$, by putting a unit in each cell where $f_1 \neq g_1$, $f_2 \neq g_2$ (Figure 22).

$$\bar{\phi} = \begin{array}{|c|c|c|c|} \hline 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 0 & 1 \\ \hline \end{array} \quad x_1, x_2 ; \; x_4, x_3$$

Figure 22.

Step 3: The solutions are determined from $x_1\bar{x}_2x_3\bar{x}_4$, $\bar{x}_1x_2x_3x_4$, $x_1x_2\bar{x}_3x_4$, i.e.,

$$\text{(a)} \quad x_1 = x_3 = 1, \quad x_2 = x_4 = 0$$

$$\text{(b)} \quad x_1 = 0 \quad x_2 = x_3 = x_4 = 1$$

$$\text{(c)} \quad x_1 = x_2 = x_4 = 1, \quad x_3 = 0$$

4.3. Pseudo-Boolean Equations

Define B_2 as the two-element Boolean algebra (i.e., the set $\{0,1\}$ plus the binary operations of logical OR (+), logical

AND $(\cdot)$, and the logical unary operation of negation or complementation $(\,\bar{}\,)$.

A pseudo-Boolean function is one that maps the cartesian product B_2^n ($= B_2 \times \ldots \times B_2$ n times) into the field of reals R, or

$$f : B_2^n \rightarrow R \tag{4.13}$$

In other words, a function of bivalent variables which assumes some real value (not necessarily 0 or 1 as in a Boolean function) for all possible inputs is a pseudo-Boolean function. Furthermore, if a one-to-one correspondence is established between the real numbers 0 and 1 and the elements of B_2, then Boolean functions are just a subset of pseudo-Boolean functions, since for every Boolean function

$$h : B_2^n \rightarrow B_2 \tag{4.14}$$

Pseudo-Boolean functions most frequently have integer values; indeed, these functions are called "integer algebraic functions" by R. Fortet [5]. The following method by P. L. Ivanescu [8] and S. Rudeanu [8] for solving simultaneous pseudo-Boolean equations might also be used to solve simultaneous Boolean equations, but it will become obvious that this would be a trivial special case. The method itself is a systematic tree-like construction and is derived from Rudeanu's method of Section 3.4.

The general form of a linear pseudo-Boolean equation is

$$a_1x_1+b_1\bar{x}_1+a_2x_2+b_2\bar{x}_2+\ldots+a_nx_n+b_n\bar{x}_n = k \qquad (4.15)$$

where a_i, $b_i (i = 1,2,\ldots,n)$ and k are given reals, and $a_i \neq b_i$ for all i. If any $a_i < 0$ or $b_i < 0$, substitute either

$$x_i = 1 - \bar{x}_i \qquad (\text{for } a_i < 0) \qquad (4.16)$$

or

$$\bar{x}_i = 1 - x_i \qquad (\text{for } b_i < 0) \qquad (4.17)$$

into (4.15) and collect terms. The i-th equation is now written in the "canonical form"

$$c^i_{i_1}\tilde{x}_{i_1} + c^i_{i_2}\tilde{x}_{i_2} + \ldots + c^i_{i_m}\tilde{x}_{i_m} = d^i \qquad (4.18)$$

where $\tilde{x} = x$ or $\bar{x}$, $x_{i_1}, \ldots, x_{i_m}$ are the variables of the i-th equation of the system (of n variables), and

$$c^i_{i_1} \geq c^i_{i_2} \geq \ldots \geq c^i_{i_m} > 0 \qquad (4.19)$$

Table 4.1

A. Equation

No.	Case	Informations		
		Conclusions	Fixed variables	Remaining equation
1^o	$d^i < 0$	No solutions	-	-
2^o	$d^i = 0$	All of appearing variables fixed	$\tilde{x}_{i_1} = \ldots = \tilde{x}_{i_m} = 0$	-
3^o	$d^i > 0$ and $c^i_{i_1} \geq \ldots c^i_{i_p} > d^i \geq$ $\geq c^i_{i_{p+1}} \geq \ldots \geq c^i_{i_m}$	Part of appearing variables fixed	$\tilde{x}_{i_1} = \ldots = \tilde{x}_{i_p} = 0$	$\sum_{j=p+1}^{m} c^i_{i_j} \tilde{x}_{i_j} = d^i$
4^o	$d^i > 0$ and $c^i_{i_1} = \ldots = c^i_{i_p} = d^i >$ $> c^i_{i_{p+1}} \geq \ldots \geq c^i_{i_m}$	There are p+1 possibilities $\alpha_1, \ldots, \alpha_p, \beta$	α_k: $\tilde{x}_{i_k} = 1$, $\tilde{x}_{i_1} = \ldots = \tilde{x}_{i_k} =$ $= \tilde{x}_{i_{k+1}} = \ldots = \tilde{x}_{i_m} = 0$ $(k=1, \ldots, p)$	-
			β: $x_{i_1} = \ldots = x_{i_p} = 0$	$\sum_{j=p+1}^{m} c^i_{i_j} \tilde{x}_{i_j} = d^i$
5^o	$d^i > 0, c^i < d^i (j=1,2,\ldots,m)$ and $\sum_{j=1}^{m} c^i_{i_j} < d^i$	No solutions	-	-

Table 4.1 (Continued)

6^o	$d^i>0, c^i_{i_j}<d^i (j=1,2,\ldots,m)$ and $\sum_{j=1}^{m} c^i_{i_j} = d^i$	All of appearing variables fixed	$\tilde{x}_{i_1} = \ldots = \tilde{x}_{i_m} = 1$	-
7^o	$d^i>0, c^i_{i_j}<d^i (j=1,2,\ldots,m)$ $\sum_{j=1}^{m} c^i_{i_j} > d^i$ and $\sum_{j=2}^{m} c^i_{i_j} < d^i$	One variable fixed	$\tilde{x}_{i_1} = 1$	$\sum_{j=2}^{m} c^i_{i_j} \tilde{x}_{i_j} = d^i - c^i_{i_1}$
8^o	$d^i>0, c^i_{i_j}<d^i (j=1,2,\ldots,m)$ $\sum_{j=1}^{m} c^i_{i_j} > d^i$ and $\sum_{j=2}^{m} c^i_{i_j} \geq d^i$	There are two possibilities δ_1, δ_2	$\delta_1 : \tilde{x}_{i_1} = 1$	$\sum_{j=2}^{m} c^i_{i_j} \tilde{x}_{i_j} = d^i - c^i_{i_1}$
			$\delta_2 : \tilde{x}_{i_1} = 0$	$\sum_{j=2}^{m} c^i_{i_j} \tilde{x}_{i_j} = d^i$

(Note that the subscript of a coefficient is not necessarily equal to that of its corresponding variable; the former denotes the position of the term in the rearranged equation, while the latter merely differentiates among the variables).

Table 4.1 below (this is table 10 from [8] p. 39) is a summary of all possible cases of (4.18) and the conclusions which may be drawn in each case.

Obviously, if one equation has no solutions or two equations result in different values for $\tilde{x}_i$, the system of equations is inconsistent. Depending on the amount of information deduced from the equations, each equation will fall into one of three classes:

1. Determinate - These are cases 1,5,2,6,3,7 in table 4.1. Case 1 and 5 are to be considered first, since they conclude that no solutions exist. Cases 2 and 6 are next considered, since in this case all of the appearing variables are fixed. Finally, in cases 3 and 7, some of the variables are fixed.
2. Partially determinate - If there are no determinate equations, those falling into case 4 would be first treated; in this case, the attack is split into p + 1 cases, each with increased information.
3. Indeterminate - This is case 8, where almost no information is available, and the attack is split into 2 cases (when all of the system equations are indeterminate).

Example: Solve the system of equations

$$-x_1+4\bar{x}_1+2x_2-2x_4+\bar{x}_4+6x_5-8x_6 = 5 \qquad (4.20.1)$$

$$x_1+x_3-3\bar{x}_3-5\bar{x}_4+x_6-\bar{x}_6+2x_7 = -1 \qquad (4.20.2)$$

$$-9\bar{x}_1-3x_2+4\bar{x}_2+4x_3+5x_6-x_7+\bar{x}_7 = -4 \qquad (4.20.3)$$

$$-3x_1+2\bar{x}_1+7\bar{x}_3-2x_4+2\bar{x}_4-\bar{x}_5+2\bar{x}_6 = 4 \qquad (4.20.4)$$

The transformations of (4.16) and (4.17) change system (4.20) to

$$8\bar{x}_6+6x_5+5\bar{x}_1+3\bar{x}_4+2x_2 = 16 \qquad (4.21.1)$$

$$5x_4+4x_3+2x_6+2x_7+x_1 = 8 \qquad (4.21.2)$$

$$9x_1+7\bar{x}_2+5x_6+4x_3+2\bar{x}_7 = 9 \qquad (4.21.3)$$

$$7\bar{x}_3+5\bar{x}_1+4\bar{x}_4+2\bar{x}_6+x_5 = 10 \qquad (4.21.4)$$

Equations (4.20) are the "canonical" form of the system. Note that (4.21.3) is the partially determinate case 4, while all others

are in the indeterminate case 8. Thus, there will be $p + 1 = 2$ possibilities: (α) $x_1 = 1$, $\bar{x}_2=x_6=x_3=\bar{x}_7=0$, and (β) $x_1 = 0$.

Case (α): $x_1=x_2=x_7=1$, $x_3=x_6=0$

This will eliminate (4.21.3) and change the other equations to

$$6x_5+3\bar{x}_4 = 6 \qquad (4.22.1)$$

$$5x_4 = 5 \qquad (4.22.2)$$

$$4\bar{x}_4+x_5 = 1 \qquad (4.22.4)$$

From (4.22.2), $x_4 = 1$, which results in $x_5 = 1$ from (4.22.1) and (4.22.4). Thus, the total solution is $x_1=x_2=x_4=x_5=x_7=1$, $x_3=x_6=0$.

Case (β): $x_1= 0$ transforms (4.21) to

$$8\bar{x}_6+6x_5+3\bar{x}_4+2x_2 = 11 \qquad (4.23.1)$$

$$5x_4+4x_3+2x_6+2x_7 = 8 \qquad (4.23.2)$$

$$7\bar{x}_2+5x_6+4x_3+2\bar{x}_7 = 9 \qquad (4.23.3)$$

$$7\bar{x}_3+4\bar{x}_4+2\bar{x}_6+x_5 = 5 \qquad (4.23.4)$$

(4.23.4) is the determinate case 3 (all others are indeterminate); consequently, let $\bar{x}_3=0$ (i.e., $x_3=1$). The system becomes

$$8\bar{x}_6+6x_5+3\bar{x}_4+2x_2 = 11 \tag{4.24.1}$$

$$5x_4+2x_6+2x_7 = 4 \tag{4.24.2}$$

$$7\bar{x}_2+5x_6+2\bar{x}_7 = 5 \tag{4.24.3}$$

$$4\bar{x}_4+2\bar{x}_6+x_5 = 5 \tag{4.24.4}$$

(4.24.2) and (4.24.3) are case 3, and (4.24.4) is case 7 (note that these cases have the same priority). Since $\tilde{x}_{i_1} = x_4$ in (4.24.2) and (4.24.4), it will be more convenient to treat one of these first, Thus, choosing $x_4 = 0$ yields the system

$$8\bar{x}_6+6x_5+2x_2 = 8 \tag{4.25.1}$$

$$2x_6+2x_7 = 4 \tag{4.25.2}$$

$$7\bar{x}_2+5x_6+2\bar{x}_7 = 5 \tag{4.25.3}$$

$$2\bar{x}_6+x_5 = 1 \tag{4.25.4}$$

(4.25.2) is the determinate case 6 which has higher priority than the other equations. Hence, let $x_6 = x_7 = 1$, yielding

$$6x_5+2x_2 = 8 \qquad (4.26.1)$$

$$7\bar{x}_2 = 0 \qquad (4.26.3)$$

$$x_5 = 1 \qquad (4.26.4)$$

The values $x_5=1$, $x_2=1$, obtained from (4.26.4) and (4.26.3) respectively, satisfy (4.26.1); therefore, the complete solution is $x_1 = x_4=0$, $x_2=x_3=x_5=x_6=x_7=1$. The families of solutions found in cases (α) and (β) are summarized in Table 4.2.

Table 4.2

x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	1	0	1	1	0	1
0	1	1	0	1	1	1

Note that there are $2^7 = 128$ possibilities for solutions, and the only two solutions were found directly with no wasted effort.

4.4. Summary and Applications

As in Section 3, the map method is the most efficient method for normal problems. Ashenhurst's method may be used effectively for very simple systems. Grigor'yan's "set-theory" method is the only one adequate to handle large systems with a great many unknowns, although a computer program may be required. The algorithm for the solution of pseudo-Boolean equations is quite compact and systematic, and is one of the most efficient methods of solution yet devised for problems of this type.

The methods for particular solutions of Boolean equations are particularly useful in the design of logic circuits, where the low and high potentials are represented by 0 and 1 levels, respectively.

Pseudo-Boolean equations are especially important in linear programming methods of operations research, used to solve all types of problems, and most frequently, those in economics and business. The solutions of these equations are also beneficial to the theory of graphs and flows in networks, as well as in switching algebra.

Ivanescu and Rudeanu extend their system to include inequalities. The procedure remains the same: obtain the system in "canonical form," make conclusions based upon determinate,

partially determinate, and indeterminate equations to reduce the system, continue the tree-like construction of the solution until all possible solutions are obtained. A special table, similar to the one for equations, is required, but by including inequalities, the power of the method is greatly increased, especially for use in linear programming. (See [8] , pp. 23-52).

5. RELATED TOPICS

The following topics are either extensions of the methods of Sections 3 and 4, or are related to the solution of simultaneous Boolean equations in a manner which engenders them as areas of future study.

5.1 Programmed Methods

In view of the amount of labor which would be required for equations more complex than those given in the examples, programmed methods should be very valuable. Grigor'yan mentions that a program has already been written for his method; moreover, since it was shown that his algorithm is analogous to Ashenhurst's, the latter method is also programmable.

The map and matrix methods use graphical devices for performing the logical expansion of equations. Since the computer would now do this, these methods would be superfluous; however, it would seem that they too could be easily programmed, and probably with greater ease than the logical algebraic methods. Certainly Ledley's methods would be of great use with a computer manipulating a finite number of designation numbers. Ivanescu and Rudeanu also mention that their technique is being programmed. Indeed, it would seem that the simultaneous solution of Boolean equations by computers might become instrumental in the design of new computers.

5.2 Simultaneous Implications

In Section 3.6, it was shown that many word-logic problems consist of propositions in the form of implications, i.e., if A, then B. Furthermore, if $A \rightarrow B$, Ledley's consequence solution stated that #B must have a unit at least in all positions where #A is a unit. Thus, Ledley's solution of general Boolean functional equations can be modified slightly to solve systems of simultaneous equivalences.

If $f_k \equiv A \rightarrow B \equiv g_k$, form the f_k and g_k matrices in the usual manner. At this point, it is more convenient to think of the remaining process in terms of Svoboda's algorithm. The map of ϕ_k will be formed by placing a 0 only where a cell in f_k is 1 and the corresponding cell in g_k is a zero; this is the result of the implication. The remainder of the solution is unchanged.

5.3 Systems Other Than Binary

Throughout this paper, all methods and problems have considered only bivalent variables, since this is the most common case. However, many of the techniques may be extended to systems of ternary variables, quaternary variables, etc. For the strict map methods, such as those of Svoboda and Nadler, the only requirement is the extension of the Veitch chart to the new bases.

For example, in a ternary system, there will be 3^n possible combinations of n variables. A sample map of three ternary variables is shown below in Figure 23. In this chart, $x=1$, $\bar{x}=2$, $\bar{\bar{x}}=0$.

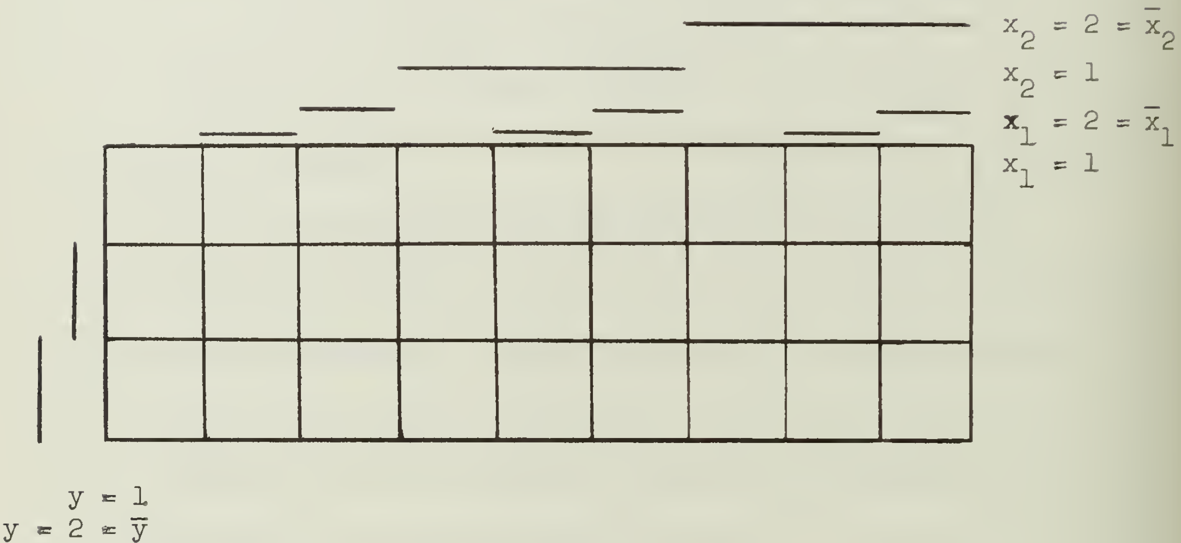


Figure 23.

It is necessary to define the logical operations of all new bases. For radix three, the logical operations of AND, OR, and NOT are defined as follows:

AND	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

OR	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

NOT	
0	1
1	2
2	0

Defining operations for logic of any base is easily accomplished by noticing that the value of a OR b is the greater of the values of a and b; also, a AND b takes the smaller value of a and b, while NOT permutes the values cyclically. Hence, for (m+1)-valued logic, the logical operations would be defined as

AND	0	1	2	3	...	m
0	0	0	0	0	...	0
1	0	1	1	1	...	1
2	0	1	2	2	...	2
3	0	1	2	3	...	3
⋮	.	.	.	.	....	.
m	0	1	2	3	...	m

OR	0	1	2	3	...	m
0	0	1	2	3	...	m
1	1	1	2	3	...	m
2	2	2	2	3	...	m
3	3	3	3	3	...	m
⋮	.	.	.	.	....	m
m	m	m	m	m	...	m

NOT	
0	1
1	2
2	3
3	4
⋮	⋮
m	0

Having defined the set of logical operations, Ledley's methods may be extended for any new base. For ternary variables, $b[x_1,x_2]$ would be

$$b\,[x_1,x_2] = \begin{cases} \#x_1 = 012 \quad 012 \quad 012 \\ \\ \#x_2 = 000 \quad 111 \quad 222 \end{cases}$$

Designation numbers are found as usual, but in this case, the new symbol, $\bar{\bar{x}}$, is possible. For example,

$$\#\bar{x}_1 = 120 \quad 120 \quad 120$$

$$\#\bar{\bar{x}}_2 = 201 \quad 201 \quad 201$$

For Type 2 problems, the solution is unchanged, except the new definitions of logical sum and product are used in the matrix multiplication. For Type 3 problems, a new kind of matrix multiplication must first be defined; this results in antecedence, consequence, and "intercedence" problems, whose solutions are of no greater difficulty than those in the binary system. Type 1 solutions may be found as before by the Θ matrix product; for (m+1)-

valued logic, an m is recorded in [ϕ] , instead of a unit, as in the binary system. (For further details, see [10], pp. 479-484.)

Extending the logical algebraic methods would simply entail the use of the new symbol $\bar{\bar{x}}$ in the equations, and expansion by means of the newly defined logical operations.

5.4 "Physical Mapping"

Instead of using a graph, such as a Veitch chart, to evaluate a function for all possible combinations of variables, Postley[16] describes a method using IBM cards. If the 12 rows and 80 columns of a card are regarded as 2 rows of 480 columns, each card could be punched to represent some combination of n ($n \leq 480$) bivalent variables.

It has been previously shown that a function may be represented as a sum of minterms. If each "type I" card represents a minterm, then column q is punched in row 1 if x_q is in the minterm, or in row 2 if $\bar{x}_q$ is in the minterm. In this manner, Type I card decks representing each equation may be compiled.

Type II cards represent the value of each of the n variables at time t. If at time t, x_k is true, row 1 in column k is punched; if $\bar{x}_k$ is true, row 2 is punched, but in either case, both rows of all other columns are punched. The deck of Type II cards for time t is called the "this time table."

If each Type I card is then sighted against the "this time table," this would be equivalent to evaluating the functions and a "next time table" may be formed. This system was devised for the sequential equations of flip-flops, but could be extended, in a manner similar to Grigor'yan's algorithm, for the determination of the particular solution of general equations.

Represent an equation as the sum of minterms, and let each minterm be represented by the number corresponding to its binary value. The first 256 positions of a card would correspond to all the possible minterms of 8 variables. By punching all positions except those corresponding to the minterms in each equation, and then sighting the deck of equation cards, the solution is simply the unpunched terms in all equations. This is equivalent to mapping $\overline{\Phi}$, and would seem to offer no advantage to a similar mapping on a Veitch chart.

5.5 Boolean Ring Equations

It has been shown that a soluable Boolean equation

$$f(x_1, x_2, \ldots, x_n) = 0 \tag{5.1}$$

may be expressed as a sum of minterms and corresponding coefficients, as in (5.2).

$$a_1x_1x_2\ldots x_n+a_2x_1x_2\ldots\bar{x}_n+\ldots+a_{2^n}\bar{x}_1\bar{x}_2\ldots\bar{x}_n = 0 \qquad (5.2)$$

This is called the "normal form" of (5.1), and the coefficients are called the "discriminants" of the equation.

A necessary and sufficient condition for the existence of a unique solution to (5.2), first obtained by Whitehead [25], is

$$a_1a_2\ldots a_{2^n} = 0 \qquad (5.3.1)$$

and

$$\bar{a}_i\bar{a}_j = 0 \quad (i \neq j) \qquad (5.3.2)$$

The elements of a Boolean algebra form a Boolean ring with respect to the operations $\oplus$ and $\cdot$. The operator " $\oplus$ " is called "ring sum" in this case, and is defined as usual:

$$a \oplus b = a\bar{b} + \bar{a}b \qquad (5.4)$$

A Boolean ring is a ring with unity in which all elements are idempotent. The following relations are true:

$$a \oplus a = 0 \qquad (5.5)$$

$$a \oplus 0 = a \qquad (5.6)$$

$$\bar{a} = 1 \oplus a \tag{5.7}$$

$$a+b = a \oplus b \oplus ab \tag{5.8}$$

$$a \oplus (b \oplus c) = (a \oplus b) \oplus c \tag{5.9}$$

$$a \oplus b = b \oplus a \tag{5.10}$$

$$a(b \oplus c) = ab \oplus ac \tag{5.11}$$

From (5.8),

$$a+b = a \oplus b \quad \text{iff } ab = 0 \tag{5.12}$$

Thus, (5.2) may be written as

$$a_1 x_1 x_2 \ldots x_n \oplus a_2 x_1 x_2 \ldots \bar{x}_n \oplus \ldots \oplus a_{2^n} \bar{x}_1 \bar{x}_2 \ldots \bar{x}_n = 0 \tag{5.13}$$

By means of (5.7), (5.13) may be written as

$$b_1 x_1 x_2 \ldots x_n \oplus b_2 x_1 x_2 \ldots x_{n-1} \oplus \ldots \oplus b_{2^n} = 0 \tag{5.14}$$

This equation is called the "0-normal ring form" of (5.1), and the b's are called the "normal coefficients." It is also easy to show that any Boolean equation may be expressed in the forms

$$\alpha_1 x_1 x_2 \ldots x_n \oplus \alpha_2 x_1 x_2 \ldots \bar{x}_n \oplus \ldots \oplus \alpha_{2^n} \bar{x}_1 \bar{x}_2 \ldots \bar{x}_n = 1 \qquad (5.15)$$

and $$\beta_1 x_1 x_2 \ldots x_n \oplus \beta_2 x_1 x_2 \ldots x_{n-1} \oplus \ldots \oplus \beta_{2^n} = 1 \qquad (5.16)$$

called the "1-normal logical form" and the "1-normal ring form," respectively.

In a lengthy paper [14], Parker and Bernstein derive numerous conditions for the unique solvability of Boolean ring equations of the forms (5.14) and (5.16). These criteria give conditions for the normal coefficients and are obtained in several ways: by analogy to the conditions of logical equations, by direct deduction of ring equations, and through the use of determinants or "complexes" (special matrices similar to determinants). No actual methods of solution are given, but the feasibility of solving systems of simultaneous Boolean ring equations is distinctly proven.

The particular solutions of simultaneous ring equations may be found by the map method of Section 4.2. Each equation is mapped on a Veitch chart, and the intersection of the maps is the required solution. The mapping is done by evaluating the equation for the

binary variables of each cell, and placing a unit in each cell which yields an identity (i.e., the equation is true). This is facilitated by considering the ring operator to denote addition modulo 2, and is made still easier if the equations are given in normal logical or normal ring form.

5.6 Sequential Equations

Methods for the general and particular solutions of simultaneous Boolean sequential equations have been demonstrated by Phister and Postley, respectively. However, the equations involved were of the most elementary type, i.e., the design equations of common flip-flops. Wang[24] presents a very thorough investigation of means of obtaining a set of explicit functions from an implicit Boolean equation.

Wang adds a sequential or time operator d to the ordinary Boolean operators such that $x \rightarrow x_t$, $dx \rightarrow x_{t+1}$, $d^2x \rightarrow x_{t+2}$, etc. Some properties of this operator are:

$$d(\overline{A}) = (\overline{dA}), \quad d(AB) = (dA)(dB), \quad d(A+B) = (dA)+(dB) \qquad (5.17)$$

$$d^k\, G(A,\ldots,B) = G(d^kA,\ldots,d^kB). \qquad (5.18)$$

In the Boolean equation,

$$H(i,\ldots,j,x,\ldots,y,d_i,\ldots,d_j,d_x,\ldots,d_y) = 0 \qquad (5.19)$$

i,....,j are input variables, and x,...,y are output variables.

The first problem investigated by Wang is to determine if

$$\begin{aligned} dx &= f(d_i,\ldots,d_j,\ i,\ldots,j,\ x,\ldots,y) \\ &\ \vdots \\ dy &= g(d_i,\ldots,d_j,i,\ldots,j,x,\ldots,y) \end{aligned} \qquad (5.20)$$

may be found. The functionals (5.20) are called "deterministic sequential functionals", and procedures are given for determining their existence, as well as obtaining them explicitly.

"Predictive sequential functionals" are effective solutions for $d_x,\ldots,d_y$ which can't be expressed as deterministic functionals. In this case, if they exist,

$$\begin{aligned} dx &= f(i,\ldots,j,x,\ldots,y,d_i,\ldots,d_j,d^2i,\ldots,d^2j,\ldots,d^qi,\ldots,d^qj) \\ &\ \vdots \\ dy &= g(i,\ldots,j,x,\ldots,y,d_i,\ldots,d_j,d^2i,\ldots,d^2j,\ldots,d^qi,\ldots,d^qj) \end{aligned} \qquad (5.21)$$

and again, procedures are given for the determination of their existence and for the explicit functionals. Deterministic and predictive sequential functionals both lead to circuit realizations, but not so the final case.

If a solution for (5.19) exists, but there is no effective solution for $d_x, \ldots, d_y$ in terms of predictive functionals, this is termed a "noneffective" solution. A procedure involving a "solution table" which determines the existence of the solution and characterizes it is given. Note that all procedures yield general parametric solutions.

LIST OF REFERENCES

[1] Ashenhurst, R. L., "Simultaneous Equations in Switching Theory," Report number BL-4, BL-S, Bell Telephone Labs. (Available from Library of Congress).

[2] Boole, G., George Boole's Collected Logical Works (1859), Volumes 1 and 2, The Open Court Publishing Company, La Salle, Illinois, 1940.

[3] Carvallo, M., Principles et Applications de l'Analyse Booléene, Gauthier-Villars, Paris, 1965, (in French).

[4] Follinger, O., and Weber, W., "Boolesche Gleichungen-Lösung und Anwendungen," Elektronische Datenverarbeitung, December, 1963, pp. 251-263, February, 1964, pp. 4-13, April, 1964, pp. 55-66, (in German).

[5] Fortet, R., "Application de l'Algèbre de Boole en Recherche Opérationnelle," Revue Francaise de Recherche Opérationnelle, Paris, Volume 4, number 14, 1964, pp. 17-26, (in French).

[6] Grigor'yan, Y., "An Algorithm for the Solution of Systems of Logical Equations," International Journal of Computer Mathematics, 1, May, 1964, pp. 51-57.

[7] Ivanescu, P., "Systems of Pseudo-Boolean Equations and Inequalities," Bull, l'Academie Polonaise des Sciences, Volume XII, Number 11, 1964, pp. 673-679.

[8] Ivanescu, P., and Rudeanu, S., Pseudo-Boolean Methods for Bivalent Programming, Springer-Verlag, Berlin-Heidelberg-New York, 1966.

[9] Klir, J., "Solution of Systems of Boolean Equations," Aplikace Matimatiky, 7, 1962, pp. 265-273, (in Czech).

[10] Ledley, R., Digital Computer and Control Engineering, McGraw-Hill Book Company, New York, 1960.

[11] Luce, R., "A Note on Boolean Matrix Theory," Proceedings of American Math. Society, Volume 3, Number 3, 1952, pp. 382-388.

[12] Maitra, K., "A Map Approach to the Solution of a Class of Boolean Equations," Communications and Electronics, Volume 59, March, 1962, pp. 34-37.

[13] Nadler, M., and Elspas, B., "The Solution of Simultaneous Boolean Equations." IRE Transactions on Circuit Theory (Correspondence), Volume CT-7, September 1960, pp. 345-346.

[14] Parker, W., and Bernstein, B., On Uniquely Solvable Boolean Equations, University of California Press, Berkeley and Los Angeles, 1955.

[15] Phister, M. Jr., Logical Design of Digital Computers, Wiley and Sons, New York, 1958.

[16] Postley, J., " A Method for the Evaluation of a System of Boolean Algebraic Equations," Math. Tables and Other Aids for Computation, Volume 9, 1955, pp. 5-8.

[17] Rouche, N., "Some Properties of Boolean Equations," IRE Transactions on Electronic Computers, Volume EC-7, Number 4, December 1958, pp. 291-298.

[18] Rudeanu, S., "Boolean Equations and Their Applications to the Study of Bridge Circuits," I. Bull. Math. de la Soc. Sci Math. Phys. de la R. P. Roumaine, 3(51), 1959, pp. 445-473.

[19] Rudeanu, S., "Irredundant Solutions of Boolêan and Pseudo-Boolean Equations, " Rev. Roumaine Math. Pures et Appl., Volume 11, Number 2, 1966, pp. 183-188.

[20] Schroeder, E., Algebra der Logik, Volume 1, Teubner Verlagsgesellschaft, Leipzig, 1890. (In German).

[21] Schubert, E., "Simultaneous Logical Equations," Communications and Electronics, Number 46, January 1960, pp. 1080-1083.

[22] Svoboda, A., "An Algorithm For Solving Boolean Equations, "IEEE Transactions on Electronic Computers, (Correspondence), Volume EC-12, October, 1963, pp. 557-559.

[23] Veitch, E., "A Chart Method for Simplifying Truth Functions," Proc., Conf. Assn. Computing Machinery, May 2-3, 1952, pp. 127-133.

[24] Wang, H., "Circuit Synthesis by Solving Sequential Boolean Equations," Zeitschrift Math. Logik und Grundlagen d. Math., Volume 5, 1959, pp. 291-322.

[25] Whitehead, A., "Memoir On the Algebra of Symbolic Logic", American Journal of Mathematics, Volume 23, 1901, pp. 134-165, 297-316.

[26] Zemanek, H., "Die Losung von Gleichungen in der Schaltalgebra", Arch. Elektr. Ubertragung, Volume 12, Number 1, January 1958, pp. 35-44. (In German).